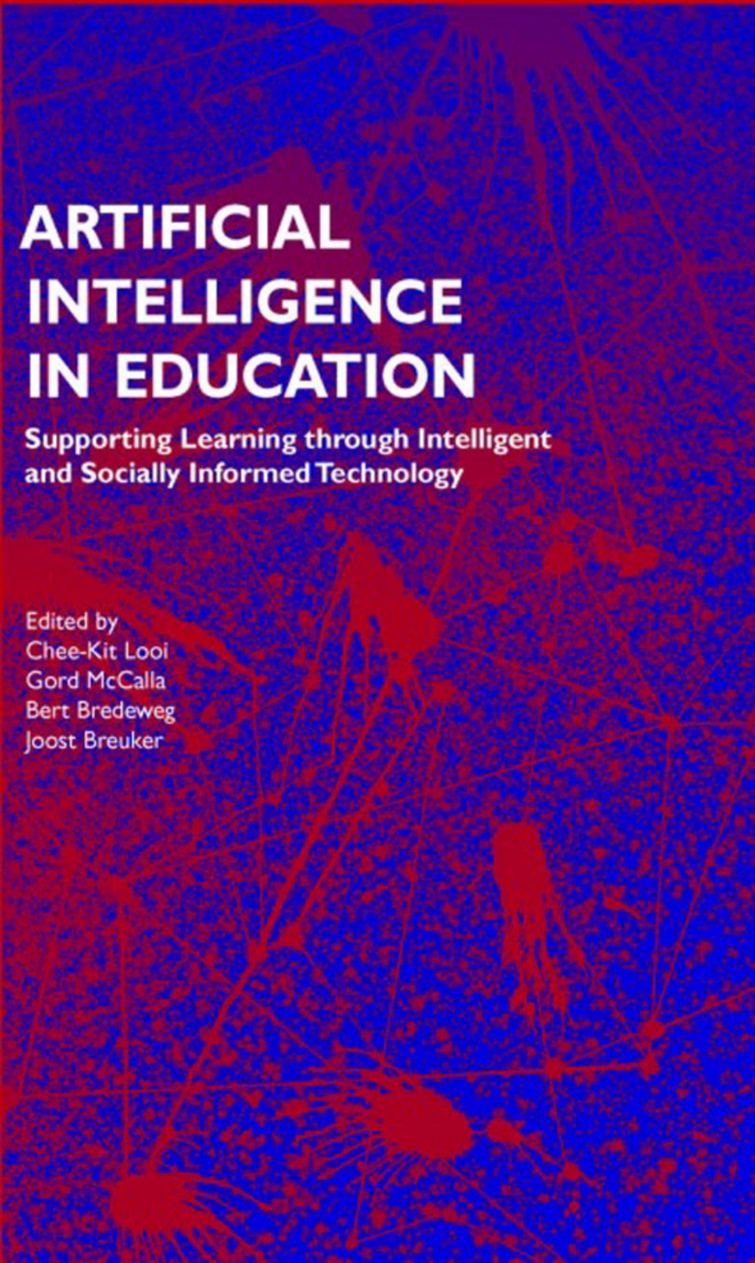
Frontiers
in
Artificial
Intelligence
and
Applications



ARTIFICIAL INTELLIGENCE IN EDUCATION

Supporting Learning through Intelligent and Socially Informed Technology

Edited by
Chee-Kit Looi
Gord McCalla
Bert Bredeweg
Joost Breuker

IOS
Press

ARTIFICIAL INTELLIGENCE IN EDUCATION

Frontiers in Artificial Intelligence and Applications

FAIA covers all aspects of theoretical and applied artificial intelligence research in the form of monographs, doctoral dissertations, textbooks, handbooks and proceedings volumes. The FAIA series contains several sub-series, including "Information Modelling and Knowledge Bases" and "Knowledge-Based Intelligent Engineering Systems". It also includes the biannual ECAI, the European Conference on Artificial Intelligence, proceedings volumes, and other ECCAI – the European Coordinating Committee on Artificial Intelligence – sponsored publications. An editorial panel of internationally well-known scholars is appointed to provide a high quality selection.

Series Editors:
J. Breuker, R. Dieng, N. Guarino, R. López de Mántaras, R. Mizoguchi, M. Musen

Volume 125

Recently published in this series

Vol. 124. T. Washio et al. (Eds.), Advances in Mining Graphs, Trees and Sequences
Vol. 123. P. Buitelaar et al. (Eds.), Ontology Learning from Text: Methods, Evaluation and Applications
Vol. 122. C. Mancini, Cinematic Hypertext –Investigating a New Paradigm
Vol. 121. Y. Kiyoki et al. (Eds.), Information Modelling and Knowledge Bases XVI
Vol. 120. T.F. Gordon (Ed.), Legal Knowledge and Information Systems – JURIX 2004: The Seventeenth Annual Conference
Vol. 119. S. Nascimento, Fuzzy Clustering via Proportional Membership Model
Vol. 118. J. Barzdins and A. Caplinskas (Eds.), Databases and Information Systems – Selected Papers from the Sixth International Baltic Conference DB&IS'2004
Vol. 117. L. Castillo et al. (Eds.), Planning, Scheduling and Constraint Satisfaction: From Theory to Practice
Vol. 116. O. Corcho, A Layered Declarative Approach to Ontology Translation with Knowledge Preservation
Vol. 115. G.E. Phillips-Wren and L.C. Jain (Eds.), Intelligent Decision Support Systems in Agent-Mediated Environments
Vol. 114. A.C. Varzi and L. Vieu (Eds.), Formal Ontology in Information Systems – Proceedings of the Third International Conference (FOIS-2004)
Vol. 113. J. Vitrià et al. (Eds.), Recent Advances in Artificial Intelligence Research and Development
Vol. 112. W. Zhang and V. Sorge (Eds.), Distributed Constraint Problem Solving and Reasoning in Multi-Agent Systems
Vol. 111. H. Fujita and V. Gruhn (Eds.), New Trends in Software Methodologies, Tools and Techniques – Proceedings of the Third SoMeT_W04

ISSN 0922-6389

Artificial Intelligence in Education

Supporting Learning through
Intelligent and Socially Informed Technology

Edited by

Chee-Kit Looi

National Institute of Education,
Nanyang Technological University, Singapore

Gord McCalla

Department of Computer Science,
University of Saskatchewan, Canada

Bert Bredeweg

Human Computer Studies,
Informatics Institute, Faculty of Science,
University of Amsterdam, The Netherlands

and

Joost Breuker

Human Computer Studies,
Informatics Institute, Faculty of Science,
University of Amsterdam, The Netherlands

IOS
Press

Amsterdam • Berlin • Oxford • Tokyo • Washington, DC

© 2005 The authors.

All rights reserved. No part of this book may be reproduced, stored in a retrieval system,
or transmitted, in any form or by any means, without prior written permission from the publisher.

ISBN 1-58603-530-4
Library of Congress Control Number: 2005928505

Publisher
IOS Press
Nieuwe Hemweg 6B
1013 BG Amsterdam
Netherlands
fax: +31 20 620 3419
e-mail: order@iospress.nl

Distributor in the UK and Ireland
IOS Press/Lavis Marketing
73 Lime Walk
Headington
Oxford OX3 7AD
England
fax: +44 1865 750079

Distributor in the USA and Canada
IOS Press, Inc.
4502 Rachael Manor Drive
Fairfax, VA 22032
USA
fax: +1 703 323 3668
e-mail: iosbooks@iospress.com

LEGAL NOTICE
The publisher is not responsible for the use which might be made of the following information.

PRINTED IN THE NETHERLANDS

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Preface

The 12th International Conference on Artificial Intelligence in Education (AIED-2005) is being held July 18–22, 2005, in Amsterdam, the beautiful Dutch city near the sea. AIED-2005 is the latest in an on-going series of biennial conferences in AIED dating back to the mid-1980's when the field emerged from a synthesis of artificial intelligence and education research. Since then, the field has continued to broaden and now includes research and researchers from many areas of technology and social science. The conference thus provides opportunities for the cross-fertilization of information and ideas from researchers in the many fields that make up this interdisciplinary research area, including artificial intelligence, other areas of computer science, cognitive science, education, learning sciences, educational technology, psychology, philosophy, sociology, anthropology, linguistics, and the many domain-specific areas for which AIED systems have been designed and built.

An explicit goal of this conference was to appeal to those researchers who share the AIED perspective that true progress in learning technology requires both deep insight into technology and also deep insight into learners, learning, and the context of learning. The 2005 theme "*Supporting Learning through Intelligent and Socially Informed Technology*" reflects this basic duality. Clearly, this theme has resonated with e-learning researchers throughout the world, since we received a record number of submissions, from researchers with a wide variety of backgrounds, but a common purpose in exploring these deep issues.

Here are some statistics. Overall, we received 289 submissions for full papers and posters. 89 of these (31%) were accepted and published as full papers, and a further 72 as posters (25%). Full papers each have been allotted 8 pages in the Proceedings; posters have been allotted 3 pages. The conference also includes 11 interactive events, 2 panels, 12 workshops, 5 tutorials, and 28 papers in the Young Researcher's Track. Each of these has been allotted a one-page abstract in the Proceedings; the workshops, tutorials, and YRT papers also have their own Proceedings, provided at the conference itself. Also in the Proceedings are brief abstracts of the talks of the four invited speakers: Daniel Schwartz of Stanford University in the U.S.A., Antonija Mitrovic of the University of Canterbury in New Zealand, Justine Cassell of Northwestern University in the U.S.A., and Ton de Jong of the University of Twente in the Netherlands.

The work to put on a conference of this size is immense. We would like to thank the many, many people who have helped to make it possible. In particular we thank the members of the Local Organizing Committee, who have strived to make sure nothing is left to chance, and to keep stressing to everybody else, especially the program co-chairs, the importance of keeping on schedule! Without their concerted efforts AIED-2005 would probably have been held in 2007! As with any quality conference, the Program Committee is critical to having a strong program. Our Program Committee was under much more stress than normal, with way more papers than expected, and a shorter time than we had originally planned for reviewing. Thanks to all of the Program Committee members for doing constructive reviews under conditions of extreme pressure, and doing so more or less on time. Thanks, too, to the reviewers who were recruited by Program Committee members to help out in this critical task. The committees organizing the other events at the conference also have helped to make the conference richer and broader: Young Researcher's Track, chaired by Monique Grandbastien; Tutorials, chaired by Jacqueline Bourdeau and Peter Wiemer-Hastings; Workshops, chaired by Joe Beck and Neil Heffernen; and Interactive Events, chaired by Lora Aroyo. Antoinette Muntjewerff chaired the conference Publicity committee, and the widespread interest in the 2005 conference is in no small measure due to her and her committee's activities. We also thank an advisory group of senior AIED researchers, an informal conference executive committee, who were a useful sounding board on many occasions during the conference planning. Each of the individuals serving in these various roles is acknowledged in the next few pages. Quite literally, without them this conference could not happen. Finally, we would like to thank Thomas Preuss who helped the program co-chairs through the mysteries of the Conference Master reviewing software.

For those who enjoyed the contributions in this Proceedings, we recommend considering joining the International Society for Artificial Intelligence in Education, an active scientific community that helps to forge on-going interactions among AIED researchers in between conferences. The Society not only sponsors the biennial conferences and the occasional smaller meetings, but also has a quality journal, the AIED Journal, and an informative web site: http://aied.inf.ed.ac.uk/aiedsoc.html.

We certainly hope that you all enjoy the AIED-2005 conference, and that you find it illuminating, entertaining, and stimulating. And, please also take some time to enjoy cosmopolitan Amsterdam.

Chee-Kit Looi, Program Co-Chair, Nanyang Technological University, Singapore
Gord McCalla, Program Co-Chair, University of Saskatchewan, Canada
Bert Bredeweg, LOC-Chair, University of Amsterdam, The Netherlands
Joost Breuker, LOC-Chair, University of Amsterdam, The Netherlands
Helen Pain, Conference Chair, University of Edinburgh, United Kingdom

International AIED Society Management Board

Paul Brna, University of Glasgow, UK – journal editor
Jim Greer, University of Saskatchewan, Canada – president elect
Riichiro Mizoguchi, Osaka University, Japan – secretary
Helen Pain, University of Edinburgh, UK – president

Executive Committee Members

Joost Breuker, University of Amsterdam, The Netherlands
Paul Brna, University of Glasgow, UK
Jim Greer, University of Saskatchewan, Canada
Susanne Lajoie, McGill University, Canada
Ana Paiva, Technical University of Lisbon, Portugal
Dan Suthers, University of Hawaii, USA
Gerardo Ayala, Puebla University, Mexico
Michael Baker, University of Lyon, France
Tak-Wai Chan, National Central University, Taiwan
Claude Frasson, University of Montreal, Canada
Ulrich Hoppe, University of Duisburg, Germany
Ken Koedinger, Carnegie Mellon University, USA
Helen Pain, University of Edinburgh, UK
Wouter van Joolingen, University of Amsterdam, Netherlands
Ivon Arroyo, University of Massachusetts, USA
Bert Bredeweg, University of Amsterdam, The Netherlands
Art Graesser, University of Memphis, USA
Lewis Johnson, University of Southern California, USA
Judy Kay, University of Sydney, Australia
Chee Kit Looi, Nanyang Technological University, Singapore
Rose Luckin, University of Sussex, UK
Tanja Mitrovic, University of Canterbury, New Zealand
Pierre Tchounikine, University of Le Mans, France

Conference Chair
Helen Pain, University of Edinburgh, United Kingdom

Program Chairs
Chee-Kit Looi, Nanyang Technological University, Singapore
Gord McCalla, University of Saskatchewan, Canada

Organising Chairs
Bert Bredeweg, University of Amsterdam, The Netherlands
Joost Breuker, University of Amsterdam, The Netherlands

Conference Executive Committee
Paul Brna, University of Glasgow, UK
Jim Greer, University of Saskatchewan, Canada
Lewis Johnson, University of Southern California, USA
Riichiro Mizoguchi, Osaka University, Japan
Helen Pain, University of Edinburgh, UK

Young Researcher's Track Chair
Monique Grandbastien, Université Henri Poincaré, France

Tutorials Chairs
Jacqueline Bourdeau, Université du Québec, Canada
Peter Wiemer-Hastings, DePaul University, United States of America

Workshops Chairs
Joe Beck, Carnegie-Mellon University, United States of America
Neil Heffernan, Worcester Polytechnic Institute, United States of America

Interactive Events Chair
Lora Aroyo, Eindhoven University of Technology, The Netherlands

Publicity Chair
Antoinette Muntjewerff, University of Amsterdam, The Netherlands

Program Committee

Esma Aimeur, Université de Montréal, Canada
Shaaron Ainsworth, University of Nottingham, United Kingdom
Fabio Akhras, Renato Archer Research Center, Brazil
Vincent Aleven, Carnegie-Mellon University, United States of America
Terry Anderson, Athabasca University, Canada
Roger Azevedo, University of Maryland, United States of America
Mike Baker, Centre National de la Recherche Scientifique, France
Nicolas Balacheff, Centre National de la Recherche Scientifique, France
Gautam Biswas, Vanderbilt University, United States of America
Bert Bredeweg, University of Amsterdam, Netherlands
Joost Breuker, University of Amsterdam, Netherlands
Peter Brusilovsky, University of Pittsburgh, United States of America
Susan Bull, University of Birmingham, United Kingdom
Isabel Fernández de Castro, University of the Basque Country UPV/EHU, Spain
Tak-Wai Chan, National Central University, Taiwan
Yam-San Chee, Nanyang Technological University, Singapore
Weiqin Chen, University of Bergen, Norway
Cristina Conati, University of British Columbia, Canada
Albert Corbett, Carnegie-Mellon University, United States of America
Vladan Devedzic, University of Belgrade, Yugoslavia
Vania Dimitrova, University of Leeds, United Kingdom
Aude Dufresne, Université de Montréal, Canada
Marc Eisenstadt, Open University,United Kingdom
Jon A. Elorriaga, University of the Basque Country, Spain
Gerhard Fischer, University of Colorado, United States of America
Elena Gaudioso, Universidad Nacional de Educacion a Distancia, Spain
Peter Goodyear, University of Sydney, Australia
Art Graesser, University of Memphis, United States of America
Barry Harper, University of Wollongong, Australia
Neil Heffernan, Worcester Polytechnic Institute, United States of America
Pentti Hietala, University of Tampere, Finland
Tsukasa Hirashima, Hiroshima University, Japan
Ulrich Hoppe, University of Duisburg, Germany
RongHuai Huang, Beijing Normal University, China
Chih-Wei Hue, National Taiwan University, Taiwan
Mitsuru Ikeda, Japan Advanced Institute of Science and Technology, Japan
Akiko Inaba, Osaka University, Japan
Lewis Johnson, University of Southern California, United States of America
David Jonassen, University of Missouri, United States of America
Wouter van Joolingen, University of Twente, Netherlands
Akihiro Kashihara, University of Electro-Communications, Japan
Judy Kay, University of Sydney, Australia
Ray Kemp, Massey University, New Zealand
Ken Koedinger, Carnegie-Mellon University, United States of America
Janet Kolodner, Georgia Institute of Technology, United States of America
Rob Koper, Open University of the Netherlands, Netherlands
Lam-For Kwok, City University of Hong Kong, Hong Kong

Susanne Lajoie, McGill University, Canada
Fong-lok Lee, Chinese University of Hong Kong, Hong Kong
Ok Hwa Lee, Chungbuk National University, Korea
James Lester, North Carolina State University, United States of America
Rose Luckin, University of Sussex, United Kingdom
Tanja Mitrovic, University of Canterbury, New Zealand
Permanand Mohan, University of the West Indies, Trinidad and Tobago
Rafael Morales, University of Northumbria at Newcastle, United Kingdom
Jack Mostow, Carnegie-Mellon University, United States of America
Tom Murray, University of New Hampshire, United States of America
Toshio Okamoto, University of Electro-Communications, Japan
Rachel Or-Bach, Emek Yezreel College, Israel
Ana Paiva, INESC-ID and Instituto Superior Técnico, Portugal
Cecile Paris, CSIRO, Australia
Peter Reimann, University of Sydney, Australia
Marta Rosatelli, Universidade Católica de Santos, Brazil
Jeremy Roschelle, SRI, United States of America
Carolyn Rosé, Carnegie-Mellon University, United States of America
Fiorella de Rosis, University of Bari, Italy
Jacobijn Sandberg, University of Amsterdam, Netherlands
Mike Sharples, University of Birmingham, United Kingdom
Raymund Sison, De La Salle University, Philippines
Amy Soller, Institute for Scientific and Technological Research, Italy
Elliot Soloway, University of Michigan, United States of America
Dan Suthers, University of Hawaii, United States of America
Erkki Suttinen, University of Joensuu, Finland
Akira Takeuchi, Kyushu Institute of Technology, Japan
Liane Tarouco, Universidade Federal do Rio Grande do Su, Brazil
Carlo Tasso, University of Udine, Italy
Pierre Tchounikine, Université du Maine, France
Kurt VanLehn, University of Pittsburgh, United States of America
Julita Vassileva, University of Saskatchewan, Canada
Felisa Verdejo, Universidad Nacional de Educacion a Distancia, Spain
Gerhard Weber, University of Trier, Germany
Barbara White, University of California at Berkeley, United States of America
Lung-Hsiang Wong, National University of Singapore, Singapore
Jin-Tan David Yang, National Kaohsiung Normal University, Taiwan
Diego Zapata-Rivera, Educational Testing Service, United States of America
Zhiting Zhu, East China Normal University, China

Reviewers

Esma Aimeur
Ainhoa Alvarez
Shaaron Ainsworth
Fabio Akhras
Vincent Aleven
Terry Anderson
Stamatina Anstopoulou
Ana Arruarte
Roger Azevedo
Mike Baker
Nicolas Balacheff
Beatriz Barros
Gautam Biswas
Bert Bredeweg
Joost Breuker
Chris Brooks
Francis Brouns
Jan van Bruggen
Peter Brusilovsky
Stefan Carmien
Valeria Carofiglio
Berardina De Carolis
Rosa Maria Carro
Isabel Fernández de Castro
Tak-Wai Chan
Ben Chang
Sung-Bin Chang
Yam-San Chee
Weiqin Chen
Yen-Hua Chen
Yu-Fen Chen
Zhi-Hong Chen
Hercy Cheng
Andrew Chiarella
Cristina Conati
Ricardo Conejo
Albert Corbett
Ben Daniel
Melissa Dawe
Yi-Chan Deng
Vladan Devedzic
Vania Dimitrova
Aude Dufresne
Hal Eden
Marc Eisenstadt
Jon A. Elorriaga
Rene van Es
Jennifer Falcone
Sonia Faremo
Bego Ferrero
Gerhard Fischer
Isaac Fung
Dragan Gasevic
Elena Gaudioso
Elisa Giaccardi
Peter Goodyear
Andrew Gorman
Art Graesser
Jim Greer
Barry Harper
Pentti Hietala
Tsukasa Hirashima
Ulrich Hoppe
Tomoya Horiguchi
RongHuai Huang
Chih-Wei Hue
Mitsuru Ikeda
Akiko Inaba
Lewis Johnson
Russell Johnson
David Jonassen
Wouter van Joolingen
Akihiro Kashihara
Judy Kay
Elizabeth Kemp
Ray Kemp
Liesbeth Kester
Ken Koedinger
Shin'ichi Konomi
Rob Koper
Yang-Ming Ku
Hidenobu Kunichika
Lam-For Kwok
Chih Hung Lai
Susanne Lajoie
Mikel Larrañaga
Fong-lok Lee
Seung Lee
Sunyoung Lee
James Lester
Chuo-Bin Lin
Fuhua Oscar Lin
Chee-Kit Looi
Susan Lu
Rose Luckin
Heather Maclaren
Montse Maritxalar
Brent Martin
Liz Masterman
Noriyuki Matsuda
Jose Ignacio Mayorga
Gord McCalla
Scott McQuiggan
Tanja Mitrovic
Frans Mofers
Permanand Mohan
Rafael Morales
Jack Mostow
Bradford Mott
Kasia Muldner
Tom Murray
Tomohiro Oda
Masaya Okada
Toshio Okamoto
Olayide Olorunleke
Ernie Ong
Rachel Or-Bach
Mourad Oussalah
Ana Paiva
Cecile Paris
Harrie Passier
Tom Patrick
Peter Reimann
Marta Rosatelli
Jeremy Roschelle
Carolyn Rosé
Fiorella de Rosis
Peter van Rosmalen
Jacobijn Sandberg
Mike Sharples
Raymund Sison
Peter Sloep
Amy Soller
Elliot Soloway
Slavi Stoyanov
Jim Sullivan
Dan Suthers
Erkki Suttinen

Akira Takeuchi
Tiffany Tang
Colin Tattersall
Pierre Tchounikine
Takanobu Umetsu
Maite Urretavizcaya
Kurt VanLehn
Julita Vassileva
Felisa Verdejo
Fred de Vries
Gerhard Weber
Barbara White
Mike Winter
Lung-Hsiang Wong
Jin-Tan David Yang
Yunwen Ye
Gee-Kin Yeo
Diego Zapata-Rivera
Zhiting Zhu

YRT Committee

Monique Baron, France
Joseph Beck, USA
Jim Greer, Canada
Erica Melis, Germany
Alessandro Micarelli, Italy
Riichiro Mizoguchi, Japan
Roger Nkambou, Canada
Jean-François Nicaud, France
Kalina Yacef, Australia

Additional YRT Reviewers

John Lee, UK
Judy Kay, Australia
Cristina Conati, Canada
Shaaron Ainsworth, UK
Peter Brusilovsky, USA
Michael Baker, France
Phil Winne, Canada
Aude Dufresne, Canada
Tom Murray, USA
Catherine Pelachaud, France

Organising Committee

Lora Aroyo, Eindhoven University of Technology, Netherlands
Anders Bouwer, University of Amsterdam, The Netherlands
Bert Bredeweg, University of Amsterdam, The Netherlands
Joost Breuker, University of Amsterdam, The Netherlands
Antoinette Muntjewerff, University of Amsterdam, The Netherlands
Radboud Winkels, University of Amsterdam, The Netherlands

Sponsors

IJAIED

The International Artificial Intelligence in Education Society

ECCAI

European Coordinating Committee for Artificial Intelligence Site

American Association for Artificial Intelligence

IOS Press

International Science Publisher

Contents

Invited Talks

Full Papers

Posters

Young Researchers Track

Panels

Tutorials

Workshops

Invited Talks

This page intentionally left blank

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Learning with Virtual Peers

Justine Cassell
Northwestern University
U.S.A.

Abstract

Schools aren't the only places people learn, and in the field of educational technology, informal learning is receiving increasing attention. In informal learning peers are of primary importance. But, how do you discover what works in peer learning? If you want to discover what peers do for one other so that you can then set up situations and technologies that maximize peer learning, where do you get your data from? You can study groups of children and hope that informal learning will happen and hope that you have a large enough sample to witness examples of each kind of peer teaching that you hope to study.

Or you can make a peer Unfortunately, the biological approach takes years, care and feeding is expensive, diary studies are out of fashion, and in any case the human subjects review board frowns on the kind of mind control that would allow one to manipulate the peer so as to provoke different learning reactions. And so, in my own research, I chose to make a bionic peer.

In this talk I describe the results from a series of studies where we manipulate a bionic peer to see the effects of various kinds of peer behavior on learning. The peer is sometimes older and sometimes younger than the learners, sometimes the same race and sometimes a different race, sometimes speaking at the same developmental level -- and in the same dialect -- and the learners, and sometimes differently. In each case we are struck by how much learning occurs when peers play, how learning appears to be potentiated by the rapport between the real and virtual child, and how many lessons we learn about the more general nature of informal learning mediated by technology.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Scaffolding inquiry learning: How much intelligence is needed and by whom?

Ton de Jong
University of Twente
The Netherlands

Abstract

Inquiry learning is way of learning in which learners act like scientists and discover a domain by employing processes such as hypothesis generation, experiment design, and data interpretation. The sequence of these learning processes and the choice for specific actions (e.g., what experiment to perform) are determined by the learners themselves. This student centeredness makes that inquiry learning heavily calls upon metacognitive processes such as planning and monitoring. These inquiry and metacognitive processes make inquiry learning a demanding task. When inquiry is combined with modelling and collaboration facilities the complexity of the learning process even increases. To make inquiry learning successful, the inquiry (and modelling and collaborative) activities need to scaffolded. Scaffolding can mean that the learning environment is structured or that learners are provided with cognitive tools for specific activities. AI techniques can be used to make scaffolds more adaptive to the learner or to developments in the learning process. In this presentation an overview of (adaptive and non-adaptive) scaffolds for inquiry learning in simulation based learning environments will be discussed.details will follow.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Constraint-based tutors: a success story

Tanja Mitrovic
University of Christchurch
New Zealand

Abstract

Constraint-based modelling (CBM) was proposed in 1992 as a way of overcoming the intractable nature of student modelling. Originally, Ohlsson viewed CBM as an approach to developing short-term student models. In this talk, I will illustrate how we have extended CBM to support both short- and long-term models, and developed methodology for using such models to make various pedagogical decisions. In particular, I will present several successful constraint-based tutors built for a various procedural and non-procedural domains. I will illustrate how constraint-based modelling supports learning and meta-cognitive skills, and present current project within the Intelligent Computer Tutoring Group.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Interactivity and Learning

Dan Schwartz
Stanford University
U.S.A.

Abstract

Two claims for artificial intelligence techniques in education are that they can increase positive interactive experiences for students, and they can enhance learning. Depending on one's preferences, the critical question might be "how do we configure interactive opportunities to optimize learning?" Alternatively, the question might be, "how do we configure learning opportunities to optimize positive interactions?" Ideally, the answers to these two questions are compatible so that desirable interactions and learning outcomes are positively correlated. But, this does not have to be the case – interactions that people deem negative might lead to learning that people deem positive, or vice versa. The question for this talk is whether there is a "sweet spot" where interactions and learning complement one another and the values we hold most important. I will offer a pair of frameworks to address this question: one for characterizing learning by the dimensions of innovation and efficiency; and one for characterizing interactivity by the dimensions of initiative and idea incorporation. I will provide empirical examples of students working with intelligent computer technologies to show how desirable outcomes in both frameworks can be correlated.

Full Papers

This page intentionally left blank

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Evaluating a Mixed-Initiative Authoring Environment: Is REDEEM for Real?

Shaaron AINSWORTH and Piers FLEMING

School of Psychology and Learning Sciences Research Institute,
University of Nottingham
Email: {sea/pff}@psychology.nottingham.ac.uk

Abstract. The REDEEM authoring tool allows teachers to create adapted learning environments for their students from existing material. Previous evaluations have shown that under experimental conditions REDEEM can significantly improve learning. The goals of this study were twofold: to explore if REDEEM could improve students' learning in real world situations and to examine if learners can share in the authoring decisions. REDEEM was used to create 10 courses from existing lectures that taught undergraduate statistics. An experimenter performed the content authoring and then created student categories and tutorial strategies that learners chose for themselves. All first-year psychology students were offered the opportunity to learn with REDEEM: 90 used REDEEM at least once but 77 did not. Students also completed a pre-test, 3 attitude questionnaires and their final exam was used as a post-test. Learning with REDEEM was associated with significantly better exam scores, and this remains true even when attempting to control for increased effort or ability of REDEEM users. Students explored a variety of categories and strategies, rating their option to choose this as moderately important. Consequently, whilst there is no direct evidence that allowing students this control enhanced performance, it seems likely that it increased uptake of the system.

1. Introduction

The REDEEM authoring tool was designed to allow teachers significant control over the learning environments with which their students learn. To achieve this goal, the authoring process and the resulting learning environments have both been simplified when compared to more conventional authoring tools. REDEEM uses canned content but delivers it in ways that teachers feel are appropriate to their learners. Specifically, material can be selected for different learners, presented in alternative sequences, with differences exercises and problems, and authors can create tutorial strategies that vary such factors as help, frequency and position of tests and degree of student control. This approach, focussing on *adapted* learning environments rather than *adaptive* learning environments, has been evaluated with respect to both the authors' and learners' experiences (see [1] for a review). Overall, REDEEM was found to be usable by authors with little technological experience and time-efficient for the conversion of existing computer-based training (CBT) into REDEEM learning environments (around 5 hours per hour of instruction). Five experimental studies have contrasted learning with REDEEM to learning with the original CBT in a variety of domains (e.g. Genetics, Computing, Radio Communication) and with a wide range of learners (schoolchildren, adults, students). REDEEM led to an average 30% improvement from pre-test to post-test, whereas CBT increased scores by 23%. This advantage for REDEEM translates into an average effect size of .51, which compares well to non-expert human individual tutors and is around .5 below full-blown ITSs (e.g. [2,3]).

To perform three of these experiments, teachers were recruited who had in-depth knowledge of the topic and the students in this class. They used this knowledge to assign different student categories which resulted in different content and tutorial strategies. In the other two experiments, this was not possible and all the participants were assigned to one category and strategy. But, it may have been more appropriate to let students choose their own approach to studying the material. This question can be set in the wider context of authoring tools research, namely for any given aspect of the learning environment, who should be making these decisions – should it be a teacher, should it be the system or can some of the authoring decisions be presented to learners in such a way that they can make these decisions for themselves. Whilst, there has been some debate in the literature about how much control to give the author versus the system [4], the issue of how much of the authoring could be performed by learners themselves has received little direct attention. Of course, the general issue of how much control to give students over aspects of their learning has been part of a long and often contentious debate (e.g. [5, 6]). There are claims for enhanced motivation [7] but mixed evidence for the effectiveness of learner control.

However, in the context under consideration (1st year University students), there was no teacher available who could make these decisions based upon personal knowledge of the student. Consequently, to take advantage of REDEEM's ability to offer adapted learning environments, the only sensible route was to allow learners to make these decisions for themselves. As a result, a mixed initiative version of REDEEM was designed that kept the same model of content and interactivity authoring as before, but now gave students the choice of learner category (from predefined categories) and teaching strategy (also predefined). Thus the aim of this approach is not to turn learners into authors as [8] but instead to renegotiate the roles of learners and authors.

A second goal for this research was to explore the effectiveness of REDEEM over extended periods, outside the context of an experiment. One positive aspect of AIED in recent years has been the increase in number of evaluations conducted in realistic contexts (e.g. [3, 9]). However, given the complex issues involved in running an experiment, the norm for evaluation (including the previous REDEEM studies) is that they are conducted in experimental situations with limited curriculum over a short duration and post-tests tend to be on the specific content of the tutor. To show that interacting with a learning environment improves performance when used as part of everyday experience is still far from common (another exception is ANDES [10] whose research goal is to explore if minimally invasive tutoring can improve learning in real world situations). Yet, it is this test that may convince sceptics about the value of ITSs and interactive learning environments. However, assessing if REDEEM improves learning 'for real' is far from easy as it was difficult to predict how many students would chose to use REDEEM or whether we would be able to account for explanations based upon differential use of REDEEM by different types of learners.

2. Brief System Description

REDEEM consists of three components: a courseware catalogue of material created externally to REDEEM, an ITS Shell and a set of authoring tools (please see [1] for a fuller description of components and the authoring process). REDEEM's authoring tools decompose the teaching process into a number of separate components. Essentially, authors are asked to add interactivity to the underlying courseware (by adding questions, hints, answer feedback and reflections points) they describe the structure of material, create student categories and create teaching strategies. This information is then combined by assigning particular teaching strategies and types of material to different learner groups. The difference with this latest version is that the students themselves select one of the learner categories and this now results in a default teaching strategy, which they can change to any other strategies that are available. This design is a trade-off between giving students' significant choice yet only requiring a minimum of interaction to utilise this functionality.

The courseware consisted of ten PowerPoint lectures saved as html. These were then imported into REDEEM by an experimenter, who in addition to describing the structure of the material, added approximately one question per page with an average of three hints per question and an explanation of the correct answer and reflection points. Four learner categories were created (non-confident learner (NCL, confident learner (CL), non-confident reviser (NCR), confident reviser (CR). Four default teaching strategies were created (Table 1) based upon ones teachers had authored in previous studies [11]. In addition, four optional strategies were devised that provided contrasting experiences such as using it in 'exam style' or in 'pre-test' mode (test me after the course, before section or course).

Table 1. Teaching Strategies

Name	**Default**	**Description**
Simple Introduction	NCL	No student control of material or questions; easy/medium questions (max one per page), 2 attempts per question, help available. Questions after page.
Guided Practice	NCR	No student control of material/questions; easy/medium questions (max one per page). 5 attempts per question, help is available. Questions after section.
Guided Discovery	CL	Choice order of sections but not questions. 5 attempts per question, help only on error. Questions after section.
Free Discovery	CR	Choice order of sections and questions. 5 attempts per question, help available
Just Browsing		Complete student control of material. No questions.
Test me after the course		No student control of material or questions. All questions at the end, 1 attempt per question, no help.
Test me before each section		Choose order of sections. Questions are given before each section. 5 attempts per question and help available on error.
Test me before the course		Student control sections All questions at the start. 5 attempts per question. Help is available.

3. Method

3.1. Design and Participants

This study employed a quasi-experimental design as students decided for themselves whether to learn with the REDEEMed lectures. All 215 first-year Psychology students (33 males and 182 females) had previously studied a prerequisite statistics course, which was assessed in the same exam as this course, but for which no REDEEM support had been available. 167 students completed both the pre-test and post-test.

3.2. Materials

Pre and post-tests were multiple-choice, in which each question had one correct and three incorrect answers. A pre-test was created which consisted of 12 multi-choice questions addressing material taught only in the first semester. Questions were selected from an existing pool of exam questions but were not completely representative as they required no calculation (the pre-test was carried out without guaranteed access to calculators). The 100 question multi-choice two hour exam was used as a post-test. These questions were a mix of factual and calculation questions. All students are required to pass this exam before continuing their studies. The experimenters were blind to this exam.

A number of questionnaires were given over the course of the semester to assess students' attitudes to studying, computers, statistics and the perceived value of REDEEM.

- A general questionnaire asked students to report on their computer use and confidence, the amount of time spent studying statistics and the desire for further support.
- An attitude to statistics questionnaire assessed statistics confidence, motivation, knowledge, skill and perceived difficulty on a five-point Likert scale.
- A REDEEM usage questionnaire asked students to report on how much they used REDEEM, to compare it to other study techniques and to rank the importance of various system features (e.g. questions, having a choice of teaching strategy).

3.3. Procedure

- All first year students received traditional statistics teaching for Semester One (ten lectures) from September to December 2003.
- Early in the second semester, during their laboratory classes, students were introduced to REDEEM and instructed in its use. They were informed that data files logging their interactions with the system would be generated and related to their exam performance but data would not passed to statistics lecturers in a way that could identify individuals. During these lessons, students were also given the pre-test and a questionnaire about their use of computers and perceptions of statistics.
- As the second semester progressed, REDEEMed lectures were made available on the School of Psychology intranet after the relevant lecture was given.
- Students logged into REDEEM, chose a lecture and a learner category. Students were free to override the default strategy and change to one of seven others at any time.
- At the end of the lecture course (the tenth lecture) another questionnaire was given to reassess the students' perceptions of statistics and REDEEM.
- Finally, two and a half weeks after the last lecture, all of the students had to complete a statistics exam as part of their course requirements.

4. Results

This study generated a vast amount of data and this paper focuses on a fundamental question, namely whether using REDEEM could be shown to impact upon learning. In order to answer this question a number of preliminary analyses needed to be carried out and criteria set, the most important being what counted as using REDEEM to study a lecture. After examining the raw data, it was concluded that a fair criterion was to say that students were considered to have studied a lecture with REDEEM if they had completed 70% of the questions for that lecture. The range of strategies allowed very different patterns of interactions, so questions answered was chosen because many students only accessed the practice questions without choosing to review the material and only one student looked at more than three pages without answering a question. Note, this criterion excludes the just browsing strategy, but this was almost never used and was no one's preferred strategy.

A second important preliminary analysis was to relate the 100 item exam to individual lectures. This was relatively simple given the relationship between the exam structure and learning objectives set by the lecturers. 42 questions were judged as assessing Semester 1 performance and so these questions provided a score on the exam that was unaffected by REDEEM. The Semester 2 questions were categorised according to the lecture in which the correct answer was covered. The 12 questions that addressed material taught in both semesters were not analysed further.

4.1. Relationship between REDEEM Use and Learning Outcomes

Table 2. Scores of REDEEM v non-REDEEM users

	Pre-test	Semester 1 Post-test	Semester 2 Post-test
REDEEM at least once (N = 90)	50.64% (15.96)	69.00% (12.08)	58.09% (13.03)
Never used REDEEM (N = 77)	49.24% (14.06)	67.32% (10.35)	53.44% (14.43)

The first analysis compared the scores of students who had never used REDEEM to those who had studied at least one lesson with REDEEM (Table 2). A [2 by 1] MANOVA on the pre-test, Semester 1 and Semester 2 scores revealed no difference for pre-test and Semester 1, but found the REDEEM users scored higher on Semester 2 (F(1,167) = 4.78, p<.03). However, this simple contrast overlooks much of the subtlety of the data. Of the 10 lectures; some students studied only 1 or 2 lectures and some all 10. Hence, the amount of REDEEM use (no. of lectures completed to 70% criterion) was correlated with exam scores (Table 3) - the more lectures studied with REDEEM, the greater the Semester 2 scores.

Table 3. Correlation between Test Scores and REDEEM use

	Pre-test scores	Semester 1 score	Semester 2 score	No. of lectures
Pre-test score		.171*	.165*	.038
Semester 1 score			.436***	.116
Semester 2 score				.287***
No. of lectures				

Note. * = p<.05, ** = p<.01, *** = p<.001 (two tailed test of significance)

A stepwise linear regression predicted the influence of REDEEM use and Semester 1 performance on Semester 2 performance. Semester 1 performance and REDEEM use combined predicted 23.7% of the variance (adjusted R squared). The model was significant (F(2, 164) = 26.814, p<.001). Beta values show that semester 1 performance (Beta = 0.415, t = 6.097, p<.001) is approximately twice as important as REDEEM use (Beta = 0.238, t = 3.50, p< .001) but both were significant predictors. Participants were predicted to do about 1% (exactly 0.954%) better for each REDEEM lecture they completed.

These analyses suggest that REDEEM improves students' performance, but it is still possible to argue that those students who used REDEEM more frequently were harder working and motivated students. A stringent test of the effectiveness of learning with REDEEM was to examine each lecture's questions on the exam individually. Furthermore, Semester 1 scores provide a good control for enhanced effort or ability. Consequently, ten ANCOVAS (partialling out Semester 1 performance) compared performance between REDEEM users (for that lecture) and non-REDEEM users (for that lecture). Performance for lectures 4, 5, 7 and 8 was significantly better for REDEEM users (F(1,179) = 9.34, p<.003; F(1,179) = 4.36, p<.04; F(1,179) = 4.26, p<.04; F(1,179) = 8.94, p<.01) (**Table 4**).

Table 4. Percentage Scores and the Number of the Questions on the Exam by Lecture

Lect.	No Ques	REDEEM users	REDEEM Non-users	Lect.	No Ques	REDEEM users	REDEEM Non-users
1	3	79.17 (23.47) N = 78	72.49 (27.33) N = 104	6	2	80.85 (30.49) N = 47	68.89 (34.48) N = 135
2	2	68.75 (33.92) N = 64	69.07 (35.16) N = 118	7	9	51.83 (19.83) N = 48	41.99 (22.54) N = 134
3	7	58.42 (19.83) N = 55	56.30 (19.47) N = 127	8	4	75.56 (25.83) N = 45	58.03 (30.61) N = 137
4	6	73.51 (22.84) N = 63	61.42 (24.90) N = 119	9	3	30.93 (30.39) N = 43	29.88 (24.25) N = 139
5	9	56.08 (16.62) N = 48	49.19 (18.49) N = 134	10	1	60.53 (49.54) N = 38	59.03 (49.35) N = 144

4.2. Student's use of REDEEM and their Perceptions of the Features Helpfulness

Participants completed questionnaires about their attitude to and experiences of computing and statistics. Consequently, we can explore if this influenced REDEEM uptake. No measure of statistical confidence, motivation or perception of statistics difficulty was related to REDEEM use (all correlations were between .11 and -.10). Similarly, no amount of computer usage or confidence influenced REDEEM usage.

Table 5. Most Commonly Chosen Category and Strategy (with Default Strategy)

Category	% Choice	Strategy	% Choice
Non confident learner	20.9%	Simple introduction (default)	21.3%
Non confident reviser	32.6%	Guided practice (default)	28.7%
Confident learner	7.0%	Guided discovery (default)	6.4%
Confident reviser	39.5%	Free discovery (default)	8.5%
		Test me before each section	6.4%
		Test me after the course	9.6%
		Test me before the course	19.1%

Students choose a learner category (and teaching strategy) for each lecture (Table 5). The choice of categories is not equal; very few students chose the category of "Confident learner". Partly as a result, few students experienced the Guided Discovery strategy. In terms of strategy, it is notable that "Confident revisers" were most likely to explore other strategies, and in particular to select "Test me before the course".

Table 6. Students who Chose Confident versus Non-Confident Categories

	Non-Confident (N=56)	Confident (N=32)
Pre-test	47.62% (15.94)	54.68% (14.65)
Semester 1	68.33% (12.21)	70.28% (9.86)
Semester 2	56.68% (12.17)	60.73% (14.37)
Confidence	1.80 (0.79)	2.37 (0.79)
Knowledge	1.98 (0.83)	2.50 (0.88)
Difficulty	1.64 (0.75)	2.28 (0.95)
Motivation	2.64 (0.99)	2.63 (1.01)

NB 2 subjects did not complete all parts of the statistics attitude questionnaire

We further analyzed whether the confidence levels expressed on statistics attitudes questionnaire related to student choice of learner category and to measures of learning. Whilst there was no relationship between reviser/learner and performance measures or attitude, but there was for confident/non-confident categories. Analysis by MANOVA (see Table 6) found that students who chose non-confident categories also judged themselves as less confident and knowledgeable on the statistics attitude questionnaire and found statistics more difficult than other subjects (F(1,86) = 9.17, p<.003; F(1,86) = 9.22, p<.003, F(1,86) = 12.03, p<.001). They also differed on their pre-test scores (F(1,86) = 4.23, p<.043) but not at post-test nor in their motivation to learn statistics.

Those students who attended the final lecture and completed the questionnaire (99 students, 60 of whom reported using REDEEM) considered REDEEM to be less useful than lectures for learning statistics, but more useful than tutorials, textbooks or working with friends. They reported they would definitely use REDEEM for the 2nd year statistics course if it was available (4.47/5) and would recommend REDEEM to next year's first years (4.45/5). Only one respondent would not use or recommend REDEEM. They ranked REDEEM's features in the following order of usefulness: Questions, Hints & Explanations of Answers, Choice of Strategy, Review facilities, Student History, and Notes tool.

5. Discussion

A number of analyses were performed to examine if use of REDEEM could be shown to impact upon exam performance. No analysis found that use of REDEEM was associated with either higher pre-test scores or higher Semester 1 scores on the post-test. However, Semester 2 scores were influenced by use of REDEEM. Those students who used REDEEM performed better and the more they studied with REDEEM the better they did. Furthermore, students whose used REDEEM more still performed better even with Semester 1 scores partialled out, which mitigates against an explanation based solely on differential use of REDEEM by motivated or higher ability students. Finally, REDEEM increased performance specifically on the exam questions that corresponded to the lectures that a student had studied with REDEEM (an average of 64% on those lecture's questions versus 54% on those they had not studied). These analyses combine to suggest that studying with REDEEM enhanced performance (1% per lecture studied), which if students complete all 10 lectures would result in an improvement of a degree class in the UK system.

Students who chose to learn with REDEEM did not differ in their attitudes to statistics, prior knowledge of statistics or attitudes to and use of computers. Studying with REDEEM does not seem to differentially attract students with different characteristics. However, their views about statistics did influence the way they used REDEEM. Students who rated themselves as less statistically confident tended to also choose the non-confident student category and also tended to have lower pre-test scores. Consequently, it would appear that students' lack of confidence about their statistics knowledge was rooted, to some extent, in insight into their understanding of statistics at the beginning of Semester 2. However, by the end of the year, these students had the same exam scores as their peers.

Students rated learning with REDEEM fairly highly. They did not see it as a substitute for lectures (nor was it intended to be), but preferred REDEEM to other forms of independent study. No doubt this was related to the provision of questions (with help and explanations) given the high rating of this feature and students' view that the statistics course should be supplemented with more questions (81%).

The second goal of this study was to explore if sharing some of the authoring decisions between student and author was helpful, consequently this version of REDEEM allowed students to choose their own learner category and strategy from pre-defined author choices. Students tended to pick revision categories rather than learning categories. This is almost certainly related to the fact that approximately 2/3rd of REDEEM use occurred after the end of term and a stunning 25% of total use was in the 36 hours prior to the exam. This also helps to explain the gradual fall in REDEEM use across lectures – many students simply started at lecture 1 and ran out of time to complete the whole course. Whilst this may not be an ideal way to learn statistics, it does show that REDEEM can provide support for students at times in which the traditional university provision is unavailable. Students were more equally split between those who chose to learn as either confident or non-confident. This choice was consistent both with their attitude to statistics and with poorer performance at the pre-test for non-confident learners.

For this study, there was no difference for the alternative categories in the sequence and structure of material (it simply replicated the original lecture), but each category had a different default teaching strategy based on previous experimental studies. Most of the students stuck with the default strategy except "Confident revisers" who rarely used the default strategy. This may indicate that students in this category had the confidence to explore the range of tutorial strategies or may also indicate that the default strategy was not appropriate. Many swapped to "Test me before the course", which is particularly interesting as generally this strategy was rated as the second least useful (4.53/7 compared to the most valued "Test me after the course" 6.55/7). This apparent contradiction can be resolved by examination of the dates of the log files which showed it was this strategy that was used increasingly in the days before the exam. This suggests that a category of “Last minute reviser” with a “Test me before the course” strategy may be a useful future option.

This study cannot reveal the contribution that choosing categories or strategies played in improving learning outcomes or enhancing uptake of this system. Nor can we determine the appropriateness of our decisions about learner categories or strategies. Students’ choice of categories does seem highly rational given the relationship to statistics attitudes and prior knowledge, and the time of year when use occurred. Students rated their opportunity to choose teaching strategies as the next most important feature after REDEEM’s question features (4.45/7). If we had used the previous version of REDEEM where authors chose the strategy it is likely we would have picked a strategy most like “Guided Discovery”. Overall, this was the least used strategy after “Just Browsing” because students rarely chose the “Confident Learner” strategy. Again we have no way of ascertaining if our choice or student’s individual choices would have resulted in better learning outcomes, but it is probable that this strategy would not have suited their last minute revision tactic.

Analysis of the data is on-going to explore how to improve the authoring of such features as questions and hints (e.g. why did studying lecture 3 not improve performance?) as well improvements to choices offered for learner category and teaching strategy. However, experiments with controlled use of the system and this quasi-experimental study suggest that learning with REDEEM is more helpful than learning without it.

6. References

[1] S. E. Ainsworth, N. Major, S. K. Grimshaw, M. Hayes, J. D. Underwood, B. Williams, and D. J. Wood, "REDEEM: Simple Intelligent Tutoring Systems From Usable Tools," in *Tools for Advanced Technology Learning Environments.*, T. Murray, S. Blessing, and S. E. Ainsworth, Eds. Amsterdam: Kluwer Academic Publishers, 2003, pp. 205-232.

[2] A. C. Graesser, N. K. Person, D. Harter, and T. T. R. Group, "Teaching Tactics and Dialog in AutoTutor," *International Journal of Artificial Intelligence in Education*, vol. 12, pp. 257-279, 2001.

[3] K. R. Koedinger, J. R. Anderson, W. H. Hadley, and M. A. Mark, "Intelligent tutoring goes to school in the big city," *International Journal of Artificial Intelligence in Education*, vol. 8, pp. 30-43, 1997.

[4] T. Murray, "Authoring intelligent tutoring systems: An analysis of the state of the art.," *International Journal of Artificial Intelligence in Education*, vol. 10, pp. 98-129, 1999.

[5] V. Aleven and K. R. Koedinger, "Limitations of student control: Do students know when they need help?," in *Intelligent Tutoring Systems: Proceedings of the 5th International Conference ITS 2000*, vol. 1839, *Lecture Notes in Computer Science*, 2000, pp. 292-303.

[6] T. C. Reeves, "Pseudoscience in computer-based instruction: The case of learner control literature," *Journal of Computer-Based Instruction*, vol. 20, pp. 39-46, 1993.

[7] T. W. Malone, "Toward a theory of intrinsically motivating instruction," *Cognitive Science*, vol. 5, pp. 333-369, 1981.

[8] I. Arroyo and B. P. Woolf, "Students in AWE: changing their role from consumers to producers of ITS content," in *Advanced Technologies for Mathematics Education Workshop. Supplementary Proceedings of the 11th International Conference on Artificial Intelligence in Education.*, 2003.

[9] P. Suraweera and A. Mitrovic, "An Intelligent Tutoring System for Entity Relationship Modeling," *International Journal of Artificial Intelligence in Education*, vol. 14, pp. 375-417, 2004.

[10] K. VanLehn, C. Lynch, L. Taylor, A. Weinstein, R. Shelgy, K. Schulze, D. Treacy, and M. Wintersgill, "Minimally invasive tutoring of complex physics problem solving," in *Proceedings of the 6th International Conference ITS 2002*, vol. 2363, S. A. Cerri, G. Gouardères, and F. Paraguaçu, Eds. Berlin: Springer-Verlag, 2002, pp. 367-376.

[11] S. E. Ainsworth and S. K. Grimshaw, "Evaluating the REDEEM authoring tool: Can teachers create effective learning environments," *International Journal of Artificial Intelligence in Education*, vol. 14, pp. 279-312, 2004.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

An architecture to combine meta-cognitive and cognitive tutoring: Pilot testing the Help Tutor

Vincent Aleven, Ido Roll, Bruce McLaren, Eun Jeong Ryu, Kenneth Koedinger
Human-Computer Interaction Institute
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh PA 15213, USA

Abstract Given the important role that meta-cognitive processes play in learning, intelligent tutoring systems should not only provide domain-specific assistance, but should also aim to help students in acquiring meta-cognitive skills. As a step toward this goal, we have constructed a Help Tutor, aimed at improving students' help-seeking skill. The Help Tutor is based on a cognitive model of students' desired help-seeking processes, as they work with a Cognitive Tutor (Aleven et al., 2004). To provide meta-cognitive tutoring in conjunction with cognitive tutoring, we designed an architecture in which the Help Tutor and a Cognitive Tutor function as independent agents, to facilitate re-use of the Help Tutor. Pilot tests with four students showed that students improved their help-seeking behavior significantly while working with the Help Tutor. The improvement could not be attributed to their becoming more familiar with the domain-specific skills being taught by the tutor. Although students reported afterwards that they welcomed feedback on their help-seeking behavior, they seemed less fond of it when actually advised to act differently while working. We discuss our plans for an experiment to evaluate the impact of the Help Tutor on students' help-seeking behavior and learning, including *future* learning, after their work with the Help Tutor.

Introduction

A number of instructional programs with a strong focus on meta-cognition have been shown to be effective, for example programs dealing with self-explanation (Bielaczyc, Pirolli, & Brown, 1995), comprehension monitoring (Palincsar & Brown, 1984), evaluating problem-solving progress (Schoenfeld, 1987), and reflective assessment (White & Frederiksen, 1998). These programs were not focused on the use of instructional software. Based on their success, one might conjecture that intelligent tutoring systems would be more effective if they focused more on the teaching of meta-cognitive skills, in addition to helping students at the domain level. A number of efforts have focused on supporting meta-cognition in intelligent tutoring systems (Aleven & Koedinger, 2002; Bunt, Conati, & Muldner, 2004; Conati & VanLehn, 2000; Gama, 2004; Luckin & Hammerton, 2002; Mitrovic, 2003). In some of these projects, the added value of supporting meta-cognition was evaluated. Aleven and Koedinger showed that having students explain their problem-solving steps led to better learning. Gama showed advantages of having students self-assess their skill level. Still, it is fair to say that ITS researchers are only beginning to evaluate the value of supporting meta-cognition in ITSs.

Our research concerns help seeking. There is evidence that help seeking is an important influence on learning (e.g., Karabenick, 1998), including some limited evidence pertaining to learning with interactive learning environments (Aleven et al., 2003; Wood & Wood, 1999). We focus on the hypothesis that an ITS that provides feedback on students' help-seeking behavior not only helps students to learn better at the domain level but also helps them to become better help seekers and thus better *future* learners. We are not aware of any experiments reported in the literature that evaluated the effect that instruction on help-seeking skill has on students' learning and their ability to become better help-seekers in the future.

In order to test this hypothesis, we have developed a Help Tutor, a plug-in tutor agent (Rich et al., 2002; Ritter, 1997) that evaluates students' help-seeking behavior and provides feedback, in the context of their work with a Cognitive Tutor (Koedinger et al., 1997). In developing such a tutor, there are a number of open issues. First, what exactly constitutes good help-seeking behavior? At one level, it seems quite clear that students should work deliberately, refrain from guessing, use the tutor's help facilities when needed and only then (for example, when a step is unfamiliar or after repeated errors), and read problem instructions and hints carefully. However, it is not always easy to know when help-seeking behavior is ineffective and detrimental to learning. For example, Wood and Wood (1999) describe a student who appeared to be requesting help from the system far too often, yet ended up with high learning gains. Furthermore, tutor development requires a detailed model that defines precisely what it means, for example, to work deliberately or to use help only when needed. The creation of such a model is a research contribution in itself. We use the model that is described in (Aleven et al., 2004). Since then it has been modified so that it captures a wider range of students' help-seeking strategies and provides feedback on only the most egregious deviations from reasonable help-seeking behavior.

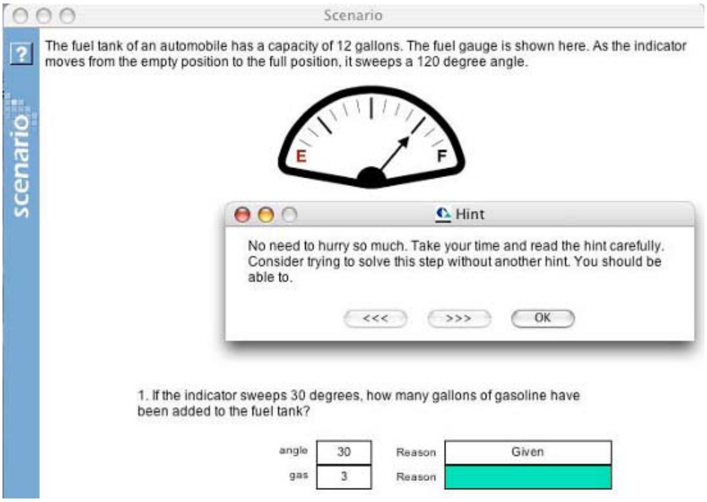
Second, how should the Help Tutor and the Cognitive Tutor be coordinated, especially when both tutors might have conflicting "opinions" about the student's action? An action can be correct on the domain level but erroneous according to the Help Tutor and vice versa. There are many coordination options, with potentially significant effect on students' learning, and very few guidelines for selecting from them. In this respect, our work has similarities to the work of Del Soldato and du Boulay (1995) whose system, MORE, coordinated the advice of a domain planner and a motivational planner. The domain planner of MORE would typically suggest that a student tackle harder problems as they succeed on easier ones, while its motivational planner might suggest repeating easier problems to improve a student's confidence and level of success.

Third, what kind of architecture can support combined cognitive and meta-cognitive tutoring? Our goal was to use the Help Tutor as a plug-in tutor agent that could be added to an existing Cognitive Tutor (or other tutoring system) with limited or no customization and, importantly, without requiring any changes to the Cognitive Tutor itself.

Although we have initial answers to these questions, we profess not to know yet if they are the right answers. Eventually, evaluation studies will have to settle that issue. There clearly is risk in our approach. Will students take the Help Tutor's advice seriously, even though it probably will not seem as directly helpful to them as the tutor's help at the domain level, to which they are accustomed? The Help Tutor must establish credibility with the students, for example, not intervene at inopportune moments, like the infamous Paper Clip. It also must not give inappropriate feedback or overly increase cognitive load. In this paper, we present our initial answers to the questions raised above and, as preliminary evidence that we are on the right track, we describe our experience pilot testing the Help Tutor with 4 students.

The Help Tutor

The Help Tutor was developed and piloted in the context of the Geometry Cognitive Tutor, an adjunct to a full-year geometry curriculum being used in approximately 350 high schools across the United States. Like all Cognitive Tutors, this tutor monitors students' step-by-step problem solutions using a cognitive model of student problem solving. It provides feedback and, at the student's request, context-sensitive hints related to the problem that the student is solving. For each problem step, multiple levels of hints are available. The hints explain which problem-solving principle applies, how it applies, and what the resulting answer is. The tutor also provides a second form of help, a searchable on-line Glossary with detailed information about the relevant geometry theorems and definitions, which students can browse freely. The tutor keeps track of the student's knowledge growth over time, using a Bayesian algorithm to estimate students' mastery of the skills targeted in the instruction (Corbett & Anderson, 1995). The Cognitive Tutors uses these estimates to select problems, while the Help Tutor uses them to determine the amount of help a student may need on any given step.



The fuel tank of an automobile has a capacity of 12 gallons. The fuel gauge is shown here. As the indicator moves from the empty position to the full position, it sweeps a 120 degree angle.

No need to hurry so much. Take your time and read the hint carefully. Consider trying to solve this step without another hint. You should be able to.

1. If the indicator sweeps 30 degrees, how many gallons of gasoline have been added to the fuel tank?

angle	30	Reason	Given
gas	3	Reason	

Figure 1: Feedback from the Help Tutor when a student abuses the tutor's context-sensitive hints

The Help Tutor is a Cognitive Tutor in its own right, built using a model of desired help-seeking behavior as a basis. This model, described in more detail in Aleven et al. (2004), is not specific to any given domain, although it is specific to the forms of assistance that Cognitive Tutors offer: feedback, context-sensitive hints, and sometimes a Glossary. According to the model, if a step in a tutor problem is familiar to the student, the student should try it. Otherwise, she should use an appropriate source of help, the Glossary on steps that are at least somewhat familiar, context-sensitive hints on unfamiliar steps. Further, the student should work deliberately: she should spend some minimum amount of time reading problem instructions and deciding what action to take. Similarly, when she requests a hint or uses the Glossary, she should spend at least some minimal amount of time with the hint or Glossary item. When she makes an error and does not know how to correct it, she should take this as a signal that she lacks the relevant knowledge and therefore should use an appropriate source of help. On the other hand, the student should not over-use the help facilities: the more familiar a step, the fewer hints she should use. Looking at too many Glossary items within a given step is also considered to be ineffective help-seeking behavior.

The model is implemented by means of 74 production rules; 36 of these rules capture productive behavior, while the remaining 38 are "bug rules" that capture unproductive behavior. The bug rules enable the Help Tutor to comment on students' unproductive help-seeking behavior, as illustrated in Figure 1. In earlier work (Aleven et al, 2004), we reported that the model identified meta-cognitive errors in 72% of student actions, when applied after the fact to an existing data set. Presenting a message to the student in so many situations is clearly not desirable. Thus, we made the model more lenient by having it focus only on the deviations most negatively correlated with learning. We also improved the model so that it estimates the minimum time it should take the student to read a hint, using research on reading rates (Card, Moran, & Newell, 1983)[1]. In implementing the model, we further had to decide how persistent the Help Tutor should be. That is, to what extent should it force students to follow its advice? For example, when recommending that the student try to solve a given step without a hint, should it withhold its hints until the student convincingly demonstrates that she

[1] A more individual-sensitive improvement we will investigate, as suggested by one of the reviewers, would be to set the minimum hint reading time based on problem solving performance, i.e., students with higher skill levels, as measured by our Bayesian algorithm, and faster problem-solving times may require less hint reading time.

is not capable of solving the step without hints? We decided not to make the Help Tutor insist in situations like this. That is, after the Help Tutor indicates that no hint may be needed, if the student repeats the hint request, the Help Tutor will not protest a second time and the requested hint will be presented. The downside of this approach is that it becomes easier for a student to ignore the Help Tutor's advice.

In integrating meta-cognitive and cognitive tutoring, there must be a way of coordinating the two tutor agents, given that there can be simultaneous, even conflicting feedback from the two sources. For instance, after a hint request by the student, the Cognitive Tutor might want to display a hint, whereas the Help Tutor might want to display a message saying that a hint is unnecessary. In principle, the two types of advice could be kept strictly separate, in space and/or time. That is, the Help Tutor's advice could be presented in a separate window or after the student completed the problem (see e.g., Ritter 1997). However, following the Cognitive Tutor principle "provide immediate feedback on errors" (Anderson et al., 1995), we decided that the Help Tutor feedback would be presented directly after a help-seeking error happens. Further, we decided that the two tutor agents would share a window in which to present messages to the student, rather than give each their own messages window. This was done to avoid the cognitive load that simultaneous messages might cause and to reduce the chance that students would miss or ignore messages from one of the agents. Conflicts between the two tutor agents are handled by a simple resolution strategy (Figure 2). First, after *answer attempts,* feedback from the Cognitive Tutor is given priority over feedback from the Help Tutor. When an answer attempt is correct from the Cognitive Tutor's point of view, it is marked as correct and no error feedback from the Help Tutor is presented, regardless of whether the student followed the desired help-seeking behavior. Coming on the heels of a successful answer, Help Tutor feedback saying, for example, that the student should have taken more time to think or should have asked for a hint instead of trying to answer, is likely to fall on deaf ears, On the other hand, when the Cognitive Tutor deems an answer attempt to be incorrect, it is flagged as incorrect. In addition, an error message may be presented from the Cognitive Tutor or from the Help Tutor. Error messages from the Cognitive Tutor are given priority, since omitting these domain-related messages may reduce the chance that the student can complete the problem. However, such messages are relatively rare in the particular Cognitive Tutor we are using. In practice, the Help Tutor messages are not overridden often.

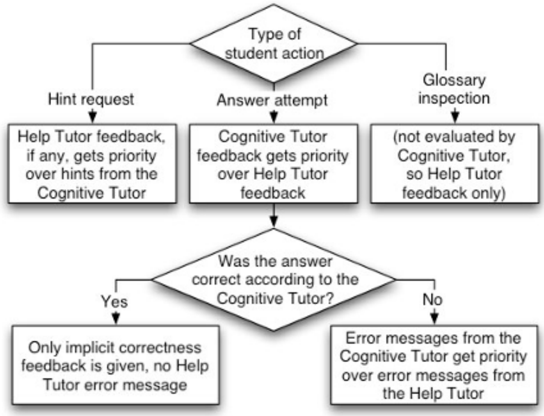

Type of student action
Hint request
Answer attempt
Glossary inspection
Help Tutor feedback, if any, gets priority over hints from the Cognitive Tutor
Cognitive Tutor feedback gets priority over Help Tutor feedback
(not evaluated by Cognitive Tutor, so Help Tutor feedback only)
Was the answer correct according to the Cognitive Tutor?
Yes
No
Only implicit correctness feedback is given, no Help Tutor error message
Error messages from the Cognitive Tutor get priority over error messages from the Help Tutor

Figure 2: Conflict resolution strategy between the two tutor agents

Second, after *hint requests*, the Help Tutor has priority. That is, if the Help Tutor deems the hint request to be inappropriate, because it is too fast or because the student should be capable of solving the step without (further) hints, the message from the Help Tutor is displayed instead of the requested hint. We hope this will turn out to be an effective way to thwart hint abuse strategies such as clicking through the hint levels at maximum speed until the last hint is reached, a way to induce the tutor to reveal the answer (documented in Aleven et al., in press). However, if the student insists and asks for more hints, the Help tutor does not block them, as discussed previously. Finally, with respect to Glossary use, there are no

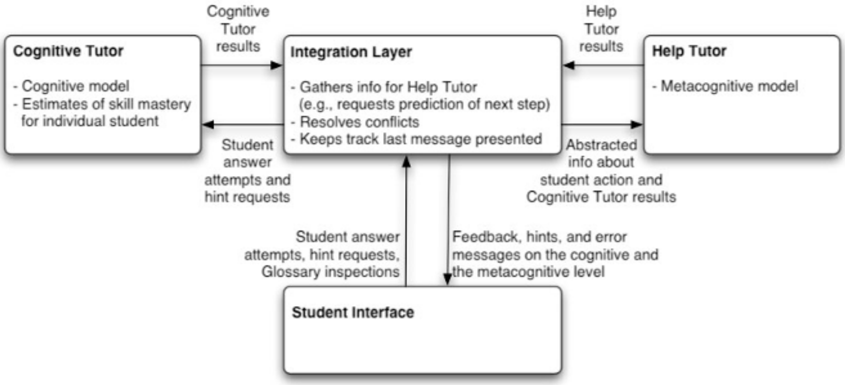

Cognitive Tutor
- Cognitive model
- Estimates of skill mastery for individual student
Cognitive Tutor results
Student answer attempts and hint requests
Integration Layer
- Gathers info for Help Tutor (e.g., requests prediction of next step)
- Resolves conflicts
- Keeps track last message presented
Help Tutor results
Abstracted info about student action and Cognitive Tutor results
Help Tutor
- Metacognitive model
Student answer attempts, hint requests, Glossary inspections
Feedback, hints, and error messages on the cognitive and the metacognitive level
Student Interface

Figure 3: Architecture with two independent tutor agents for combined cognitive and meta-cognitive tutoring

coordination issues, since the Cognitive Tutor does not evaluate students' actions with the Glossary. (Only the Help Tutor does.)

A two-agent architecture

Our goal in developing the Help Tutor was to make it an independent plug-in agent that could be added to existing Cognitive Tutors with little or no customization and without changing the Cognitive Tutor. We realized this objective in a manner similar to the multi-agent approach proposed in Ritter (1997), in which multiple tutor agents are combined in such a way that they maintain their independence. Not only is such modular design good software engineering practice, it is also necessary if the tutor agents are to be easily re-usable. A separate mediator module coordinates the tutor agents. One would typically expect this mediator to be specific to the particular set of tutor agents being combined.

Our architecture, shown in Figure 3, includes two tutor agents: a domain-specific Cognitive Tutor (i.e., an existing tutor, without modifications) and a domain-unspecific Help Tutor. Each of these tutor agents has an identical architecture, the regular Cognitive Tutor architecture, in which a cognitive model is used for model tracing – only their cognitive model is different. An Integration Layer makes sure that the Help Tutor receives all information it needs about the student's interaction with the Cognitive Tutor and resolves conflicts between the two tutor agents in the manner described in the previous section.

In order to evaluate a student's action from the perspective of help seeking, the Help Tutor needs only an abstract characterization of that action, without any domain-specific information, most importantly, the type of the action (attempt at solving a step, hint request, or Glossary lookup), its duration, the student's estimated level of mastery for the skill involved in the step, and, if the action is an attempt at answering, the Cognitive Tutor's evaluation of its correctness. Most of this information is produced in the normal course of business of a Cognitive Tutor. However, some information is needed earlier than it would normally be available, adding to the complexity of the Integration Layer. For example, in order to relate a student's Glossary browsing actions to an appropriate step in the problem, it is sometimes necessary to predict what step the student will work on next, before the student actually attempts that step. To do so, the Cognitive Tutor's model of geometry problem solving is cycled behind the scenes, invisible to the student. The Integration Layer has a number of additional, somewhat mundane, responsibilities, for example, to make sure that the Help Tutor knows which hint or feedback message the student is looking at (i.e., one from the Help Tutor or the Cognitive Tutor), so that it can estimate a minimum reading time. It also makes sure that hint sequences that were interrupted by Help Tutor feedback are resumed at the point of interruption, when the student issues an additional hint request. Such human-computer interaction aspects, we believe, will be an important factor influencing the students' acceptance of the Help Tutor.

A pilot study with the Help Tutor

So far we have evaluated the Help Tutor using existing log files of student-tutor interactions (Aleven et al., 2004). That activity helped in validating the model, but did not produce any information about how students react to its advice. Therefore, we conducted a small-scale pilot study to find out (a) whether students perceive the Help Tutor in a positive light, (b) whether and how the Help Tutor influences their behavior, and (c) whether the Help Tutor intervenes with appropriate frequency. Four high-school students from a public school in a suburban area worked with the Help Tutor. Three of them worked with the tutor for two sessions, one week apart. The fourth student worked with the tutor for the second session only. The students were accustomed to working with the Geometry Cognitive Tutor, as they use it regularly in their classroom, but they were not familiar with the particular curriculum unit involved in the study. The Help Tutor sessions took place during class periods during which the students normally used the Cognitive Tutor, but in a different classroom, separate from the other students in the class, who did not participate in the pilot study. The Help Tutor was modified between the sessions, to fix some problems that were detected during the first session, (mainly usability problems), either by making changes to the model of desired help-seeking behavior or to the Integration Layer.

The results presented here relate to the second session only. Students completed a total of 685 actions (defined as answer attempts, hint requests, or Glossary inspections). The overall ratio of help-seeking errors (according to the Help Tutor) was 16%, ranging from 9% to 24% for the different students (see Table 1). This frequency seems reasonable, since it means that the Help Tutor intervenes once for every six student actions. It suggests that we were successful in making the model a more useful (lenient) standard for help-seeking behavior. (As mentioned above, in an earlier study involving an earlier version of the model, 72% of student actions deviated from the model.) Even more encouraging was the fact that the rate of help-seeking errors dropped from 18% during the first half of the sessions to 14% during the second half. A decrease was observed for all students. These results are only preliminary, as discussed further below. Still, the reduction in error rate is statistically significant (paired-t=4.0, p<0.03), evidence that the students adapted their behavior to the tutor. Interestingly, the reduction in the error rate cannot be attributed to the students' getting more fluent with the geometry material, since it occurred irrespective of the student's skill level for the given step (high skill: from 16% to 10%; low skill: from 33% to 29%). These numbers are based on the same definition for high/low skill as the Help Tutor uses when evaluating students' help-seeking actions, which in turn are based on the Cognitive Tutor's estimates of skill mastery. Particularly noteworthy is the reduction in errors related to students' help requests, such as asking for hints rapidly and repeatedly. The error-rate for hint requests dropped from 43% during the first half of the students' sessions to 20% during the second half. Previously we found that this behavior is significantly negatively correlated with learning gains and is the most common help-seeking bug (Aleven et al., 2004). Therefore, reducing it was an important goal in building the Help Tutor.

At the end of each session, the students filled out a questionnaire in which they were asked whether they welcomed tutor feedback suggesting that they work slower, ask for a hint, or try without using a hint. They were asked also whether the tutor made these suggestions at appropriate times and with reasonable frequency. One of the four students, though being fond of the Help Tutor after the first session, was quite annoyed by it after the second. She did not like the tutor's suggestions that she reduce the number of hint requests. During the two sessions, this student received more than twice the number of error messages following her hint requests than the other students, due to her faulty use of help. The other three students had

Table 1: Frequency of help-seeking errors during the pilot study

% errors	Student 1	student2	student 3	Student 4	Overall
1st half	20%	27%	10%	15%	18%
2nd half	18%	21%	7%	12%	14%
Overall	19%	24%	9%	13%	16%

a positive opinion about the tutor. All three wanted the tutor to offer suggestions that they work slower and they thought that the tutor presented them at appropriate moments. Two of the three welcomed suggestions from the tutor that they try a step by themselves and thought the tutor presented them with appropriate frequency. The third student thought that these messages are unnecessary.

All in all, these answers are encouraging. They seem to indicate that the Help Tutor's advice was perceived as appropriate and that the Help Tutor did establish some credibility with the students. This is not to say that they always reacted positively at the moment that they received feedback from the Help Tutor. Particularly the "try by yourself" messages were not very popular, as they made it harder for students to get hints. After such a message, one student said: "I hate this tutor!" and another replied: "Because it makes you do the work yourself…" Such comments should probably not be taken as a sign that the tutor was ineffective. It is not unusual for students to complain when working with Cognitive Tutors, even though on the whole, there is clear evidence that the tutors are motivating (Schofield, 1995). Furthermore, if the Help Tutor makes students work harder and does so in an appropriate manner, that may well have a positive influence on students' learning outcomes.

Conclusion

We report on research to investigate whether intelligent tutoring systems can be made more effective if they provide meta-cognitive tutoring, in addition to domain-level tutoring. Our effort is different from other projects in that it focuses on a different meta-cognitive skill, help seeking, and moreover, we focus on *tutoring* a meta-cognitive skill, rather than *scaffolding* it. A key difference is that we do not try to prevent help-seeking errors, but rather, provide feedback when they occur, which we believe will be more effective in getting students to assimilate effective strategies that can and should be used in learning in general.

In developing the Help Tutor, we wanted to make sure that it is a re-usable component that can be plugged in to existing tutors with little or no customization. We achieved this goal by means of an architecture that includes a Cognitive Tutor and Help Tutor as independent agents. This architecture will facilitate the re-use of the Help Tutor in different tutor units and tutors. For example, while we initially implemented the Help Tutor in the Angles unit of the Geometry Cognitive Tutor we are now using it in the Circles unit. This transition was very smooth. In order to use the Help Tutor in conjunction with other units, such as the Similar Triangles unit, some customization will be necessary, due to extra optional tools that students can use in these units, but we do not expect that it will be very burdensome to do so.

The results from a pilot study with the Help Tutor, involving four students, are cause for cautious optimism. The students seemed to adapt to the Help Tutor, as suggested by the fact that over the limited time that they used the Help Tutor, their meta-cognitive error rate went down. Further, in their questionnaires, three of the four students reported that they welcomed the Help Tutor's input and that they found that the Help Tutor gave appropriate feedback. Thus, the Help Tutor seemed to have established some credibility in the eyes of these students. However, these results should be treated with caution. The pilot study was of short duration, involved only a small number of students, and took place outside the real classroom context –in the school itself, during regular Cognitive Tutor lab time, but in a separate room.

We are now conducting a controlled experiment to evaluate the impact of the Help Tutor when it is used in an actual classroom over an extended period of time. This experiment will address key questions that the pilot study left unanswered, such as the Help Tutor's effect on students' learning outcomes and whether it helps them to become better *future* learners.

Acknowledgements

We would like to thank Matthew Welch and Michele O'Farrell for their assistance. This research is sponsored by NSF Award IIS-0308200 and NSF Award SBE-0354420 to the Pittsburgh Sciences of Learning Center. The contents of the paper are solely the responsibility of the authors and do not necessarily represent the official views of the NSF.

References

Aleven, V., & Koedinger, K. R. (2000). Limitations of Student Control: Do Student Know when they need help? In *Proceedings of the 5th International Conference on Intelligent Tutoring Systems, ITS 2000,* 292-303, Berlin: Springer Verlag

Aleven, V., McLaren, B. M., & Koedinger, K. R. (to appear). Towards Computer-Based Tutoring of Help-Seeking Skills. In S. Karabenick & R. Newman (Eds.), *Help Seeking in Academic Settings: Goals, Groups, and Contexts*. Mahwah, NJ:Erlbaum.

Aleven, V., McLaren, B., Roll, I. & Koedinger, K. R. (2004). Toward tutoring help seeking - Applying cognitive modeling to meta-cognitive skills. In *Proceedings of the 7th International Conference on Intelligent Tutoring Systems, ITS 2004,* 227-239, Berlin: Springer Verlag.

Aleven, V., Stahl, E., Schworm, S., Fischer, F., & Wallace, R. M. (2003). Help seeking and help design in interactive learning environments. *Review of Educational Research, 73*(2), 277-320.

Card, S., Moran, T, & Newell, A. (1983). *The Psychology of Human-Computer Interaction.* Mahwah, NJ: Erlbaum.

Conati C. & VanLehn K. (2000). Toward computer-based support of meta-cognitive skills: A computational framework to coach self-explanation. *International Journal of Artificial Intelligence in Education, 11,* 398-415.

Corbett, A.T., Anderson, J.R. (1995) Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *User Modeling and User-Adapted Interaction, 4,* 253-278.

Del Soldato, T. & du Boulay, B. (1995). Implementation of motivational tactics in tutoring systems. *International Journal of Artificial Intelligence in Education, 6,* 337-378.

Gama, C. (2004). Meta-cognition in Interactive Learning Environments: The Reflection Assistant Model. In *Proceedings 7th Intern. Conf. on Intelligent Tutoring Systems, ITS 2004* (pp. 668-677). Berlin: Springer.

Gross, A. E., & McMullen, P. A. (1983). Models of the help-seeking process. In J. D. Fisher, N. Nadler & B. M. DePaulo (Eds.), *New directions in helping* (Vol. 2, pp. 45-61). New York: Academic Press.

Karabenick, 1998. *Strategic help seeking: Implications for learning and teaching.* Mahwah, NJ: Erlbaum.

Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Intelligence in Education, 8,* 30–43.

Luckin, R., & Hammerton, L. (2002). Getting to know me: Helping learners understand their own learning needs through meta-cognitive scaffolding. In *Proceedings of Sixth International Conference on Intelligent Tutoring Systems, ITS 2002* (pp. 759-771). Berlin: Springer.

Renkl, A. (2002). Learning from worked-out examples: Instructional explanation supplement self-explanations. *Learning & Instruction, 12,* 529-556.

Rich, C., Lesh, N.B., Rickel, J. & Garland, A. (2002). A Plug-in Architecture for Generating Collaborative Agent Responses, In *Proceedings of the International Joint Conference on Autonomous Agents and Multi-Agent Systems, AAMAS* (pp. 782-789).

Ritter, S., 1997. Communication, Cooperation and Competition among Multiple Tutor Agents. In B. du Boulay & R. Mizoguchi (Eds.), *Artificial Intelligence in Education, Proceedings of AI-ED 97 World Conference* (pp. 31-38). Amsterdam: IOS Press.

Roll, I., Aleven, V., & Koedinger, K. (2004), Promoting Effective Help-Seeking Behavior through Declarative Instruction. In *Proceedings of the 7th International Conference on Intelligent Tutoring Systems, ITS 2004,* 857-859. Berlin: Springer Verlag.

White, B., & Frederiksen, J. (1998). Inquiry, modeling, and meta-cognition: Making science accessible to all students. *Cognition and Instruction, 16*(1), 3-117

Wood, H., & Wood, D. (1999). Help seeking, learning and contingent tutoring. *Computers and Education, 33,* 153-169.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

"à la" in Education: Keywords Linking Method for Selecting Web Resources

Mirjana ANDRIC*, Vladan DEVEDZIC**, Wendy HALL*, Leslie CARR*
**Intelligence, Agents, Multimedia Group, School of Electronics and Computer Science,*
University of Southampton, Southampton SO17 1BJ, UK
{ma00r, wh, lac}@ecs.soton.ac.uk
***FON – School of Business Administration, University of Belgrade*
Jove Ilića 154, POB 52, 11000 Belgrade, Serbia and Montenegro
devedzic@galeb.etf.bg.ac.yu

Abstract. The authors of the Web-based courseware typically face problems such as how to locate, select and semantically relate suitable learning resources. As the concept of the Semantic Web has not yet matured, the authors resort to a keyword-based search and bookmarking. This paper proposes a tool that supports the authors in their tasks of selection and grouping the learning material. The "à la" (Associative Linking of Attributes) in Education, enhances the search engine results by extracting the attributes (keywords and document formats) from the text. The relationships between the attributes are established and visualised in a novel hypertext paradigm using the ZigZag principles. Browsing the related metadata provides a quick summary of the document that can help in faster determining its relevancy. Also, the proposed solution enables better understanding why some resources are grouped together as well as providing suggestions for the further search. The results of a user trial indicate high levels of user satisfaction and effectiveness.

Keywords. Web-based teaching, Authoring support tools, Extracting metadata, Hypertext structures, ZigZag, zzstructures.

1. Introduction

1.1 Authoring Web-based courseware

Web-based education has become a very important branch of educational technology. For learners, it provides access to information and knowledge sources that are practically unlimited, enabling a number of opportunities for personalised learning, tele-learning, distance-learning, and collaboration, with clear advantages of classroom independence and platform independence. On the other hand, teachers and authors of educational material can use numerous possibilities for Web-based course offering and teleteaching, availability of authoring tools for developing Web-based courseware, and cheap and efficient storage and distribution of course materials, hyperlinks to suggested readings, digital libraries, and other sources of references relevant for the course

In the context of Web-based education, educational material is generally distributed over a number of *educational servers*, Figure 1 [5]. The authors (teachers) create, store, modify, and update the material working with an *authoring tool* on the client side.

In a typical scenario of creating the learning material in such a context, the author would browse a number of educational servers and look for other resources on the Web. Then (s)he would reuse and reorganise parts of the material found, creating a new learning material. Generally, the new material will take the form of a sequence or a network of interconnected learning objects. Some typical problems that may arise in this scenario are:

- How to locate suitable learning resources on the Web?
- How to select the most appropriate resources for further reuse in composing the new learning material to suite the learners' needs?
- How to effectively correlate selected resources and create groups of semantically related resources to be used in the next step of creating the new material?

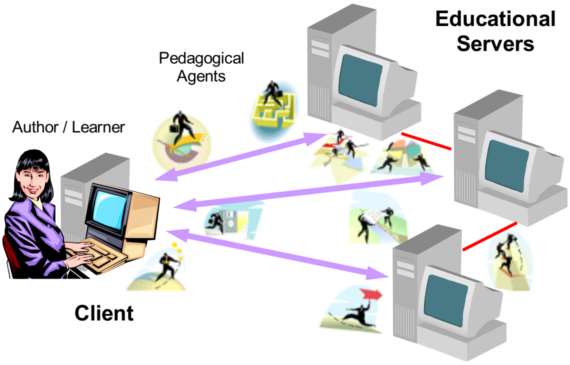


Figure 1. The context of Web-based education (after [5])

With the current technology, the author typically uses a search engine to locate the learning material on the Web. One drawback is that it is a keywords-based search, since the metadata by which the educational content on the Web can be classified is still largely lacking. Although there are advances to this end in the area of the Semantic Web [6], it is not commonplace yet. Moreover, in order to select a resource (find out whether it is relevant or not), the author must read it through. If (s)he prefers to store the reference to the resource for future use, it results in individual bookmarking and creates another typical classification problem – to remember what Web pages were similar and for which reason.

1.2 Summary of the proposed solution

The solution to the stated problem, proposed in this paper, builds on top of the existing solution consisting of search engine usage. In our “à la” (Associative Linking of Attributes) method [1], the results obtained using the search engines solution are enhanced by post-processing. In essence, search engine results are retrieved and the attributes, mainly keywords, are extracted from the textual resources. Then, the relationships between the attributes are statistically analysed and established. Subsequently, the attribute connections are visualised in a novel hypertext paradigm using the ZigZag [9] principles. The author is able to browse the keywords and their links and to select the most promising documents. Finally, selected documents and their keywords are saved into a document collection, ready for later browsing and amending. This solution seems to be more promising than purely the use of a the search engine because:

- It enables better understanding of why the resources are similar i.e. which keywords do they share;
- It provides a set of keywords acting as a summary of the web document, which enables easier selection of the relevant ones;
- Finally, it provides suggestions of the keywords to further search by.

The prototype system was built in order to investigate the research ideas. The system was evaluated in a user trial in which a set of 20 teachers were trying to sequence a web based course with and without the “à la” system. The results obtained using a post-trial questionnaire and the Wilcoxon statistical test, indicate the higher level of the user satisfaction and effectiveness, compared to the standard, search-engine only, solution.

2. Background and the Related Work

2.1 Ontologies and the Semantic Web in AIED

An important related research area in AIED is that of ontologies and ontology-aware authoring tools [2, 3, 6]. Ontologies enable machine understanding and machine processing of different contents. In education, they standardise and provide interpretations for educational contents on the Web. However, making the contents understandable and processable by pedagogical agents requires the corresponding educational Web pages to contain semantic markup, i.e. descriptions which use the terminology that one or more ontologies define and contain pointers to the network of ontologies. Using ontologies as references in marking-up educational pages on the Semantic Web enables knowledge-based indexing and retrieval of services by pedagogical agents and humans alike.

Ontology-aware authoring tools assist the authors in structuring the domain, the courseware, and the topics to be presented to the learner. They use domain ontologies as firm and stable knowledge backbone to which the author can attach the courseware material. Domain visualisation techniques help ontology-aware authoring tools communicate ontological knowledge to the author more effectively. Moreover, these tools guide the authoring process by providing the ontology of authoring tasks.

The only problem with ontologies is that they are still relatively sparse. A vast majority of domains and topics are still not supported by appropriate ontologies, which makes it difficult for authors to use ontological support.

2.2 Web Mining

Another growing branch of related research is Web mining for learning resources (e.g. see [11]). The area of Web mining relevant for the topic of this paper is called Web content mining. It refers to deploying personalised, ontology-enabled pedagogical agents to continuously go collect globally distributed content and knowledge from the Web (large Web data repositories such as documents, logs, and services) and organise it into educational Web servers [6]. The collected data can then be incorporated with locally operational knowledge bases and databases to provide a dedicated community of learners with centralised, adaptable, intelligent Web services.

2.3 ZigZag

ZigZag represents an innovative information storing paradigm introduced by a hypertext pioneer T. Nelson [9, 10]. The idea of using a complex matrix-like structure for storing and manipulating pieces of information in multiple contexts “may be thought of as a multidimensional generalization of rows and columns, without any shape or structure imposed” [9]. The data stored in the elementary cells of this matrix-like structure, known as *zzstructure*, are connected using untyped links. The cells can be connected to each other along an unlimited number of dimensions, which effectively represent types of relationships. It is convenient to colour the links belonging to the same dimension with the same colour.

However, the way of connecting cells in the zzstructure conforms to a limitation, known as a *restriction R* [8]. A cell can participate in a dimension by connecting to the other cell(s) (or in the special case to itself) via two poles, positive and negative. A restriction R says that for a particular dimension, a cell can be connected in such a way that none, one or both of the poles are used. Therefore, there exists a constraint that a cell can have at most one neighbour on each side. In other words, cells are connected in a series of sequences or lists if only one dimension is observed at a time. This collection of strands or paths, in zzstructure called ranks, form a network, effectively, an edge-coloured directed multigraph subject to *restriction R*, explained above [8]. The zzstructure principle allows an interesting effect of criss-crossing lists, where a cell can exist on many lists at the same time [9]. The structure is in general extremely difficult to visualise. Usually, only a portion of the whole structure is shown at a time, typically revealing cells and links in 2 or 3 dimensions.

2.4 London Tube: zzstructure Example

An excellent example of a zzstructure is a system of the underground train lines and stations. Stations represent cells while the train lines can be considered as dimensions. Some stations can belong to more than one line, where different ranks intersect. Moreover, in the example of the London tube system given on Figure 2, each line is given a name and a specific colour. A traveller on the network can follow some rank or change the dimension on a certain cell/station, providing that such cell offers a choice of interconnection.

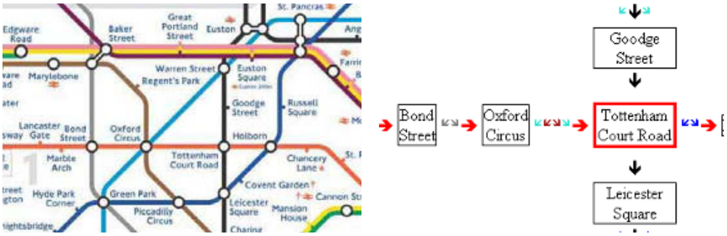


Figure 2. Portion of the London underground network on a map[1] and in the ZigZag Browser [4]

The diagram on the right of Figure 2, provides a view on the zzstructure with a Southampton University developed ZigZag Browser [4]. The cell in a ZigZag browser is represented with a rectangle, while links are represented as arrows. As it can be seen on Figure 2, some cells have several links indicated with slanted arrows. A traveller at the Tottenham Court Road station can decide to continue left, following the red-coloured, *Central line* towards *Oxford Circus*, or to change the dimension/line to a black-coloured, *Northern Line,* and go down to *Leicester Square.*

3. The “à la” Platform for Education

3.1 System Overview

The idea of presenting the interconnected pieces of the information, in fact the simple ontology network, in zzstructures, has been an inspiration for the “à la” system [1]. The central idea of the “à la” method for education is that extracting some metadata (or attributes) from the Web textual resources, analysing their relationships and storing them into a zzstructure, which is

[1] www.thetube.com

later browsed, can improve the process of searching and selecting the learning material on the Web.

In order to achieve the set goal, the system needs to perform the three main steps:

- Building the attributes-links network;
- Providing the user with a browser tool for this network;
- Selecting and saving references (URLs) and attributes of chosen web documents.

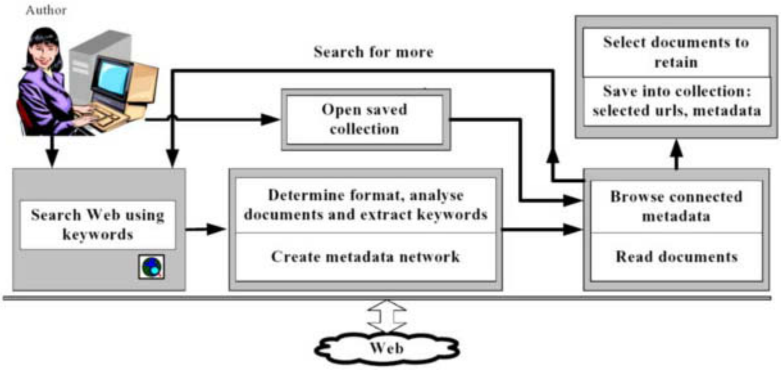

Author
Search for more
Select documents to retain
Save into collection: selected urls, metadata
Open saved collection
Search Web using keywords
Determine format, analyse documents and extract keywords
Create metadata network
Browse connected metadata
Read documents
Web

Figure 3. The "à la" Platform for Education: Block Architecture

The "à la" platform for education architecture is presented on Figure 3. The course author can use this enriched search system either by posting a regular query to a search engine or opening a previously saved pre-processed document collection. In the first case, a set of keywords is sent to a search engine (for example Google) and the results analysed. Two types of attributes are harvested: a file format and a set of keywords, using a TF-IDF machine learning technique [7]. Then, the algorithm for creating the metadata network builds an attribute network and stores it to a zzstructure, which is later presented to the user. In the second case the user opens the attribute network previously saved in a collection. The user can then browse the attribute network, familiarising her/himself with the keywords, formats and with an information about which ones of them appear in which documents. From that moment, the user can:

- Decide to read the content of some document if its keywords or links to other documents appear to be of the interest;
- Decide to use the browsed keywords in order to expand or replace the old search terms and then ask for more search engine results;
- Select the interesting documents and save the whole structure in a named document collection for the later usage.

*3.2 The "*à *la" Implementation Highlights*

The "à la" method for education uses a very simple set of attributes and relationships for building its metadata network. Only two types of metadata are considered: the Web *document format* (such as HTM or PDF) and the *keyword*, meaning the term that is among the most frequent terms in the text. This set of metadata is chosen because it is available on the Web in most of the cases. The attributes and the relationships in which they participate are shown on Figure 4. Note that there exists a relationship for each direction, as for example a document can contain many keywords, while a keyword can appear in many documents.

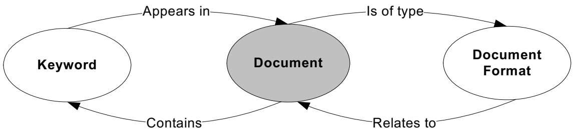
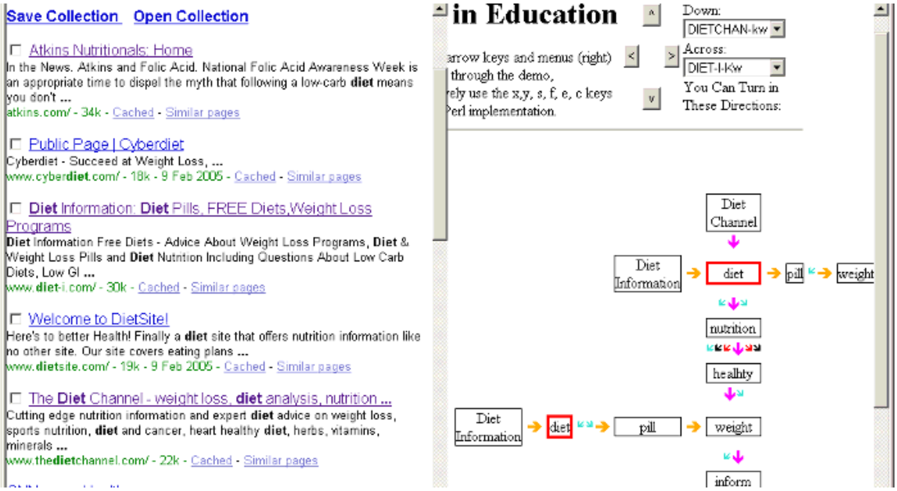


Figure 4. Types of attribute links i.e. relationships analysed in the "à la" system

In the "à la" method, attributes are firstly extracted and the keywords are stemmed. The attribute values, the actual instances of the keywords, document titles or document formats, become unique cells in a zzstructure. Each of the four relationships becomes a dimension. Subsequently, each of the values is analysed and its links established with the appropriate cells on a given dimension. For example, a document becomes connected to an array of its ordered keywords (by frequency) on the dimension called *Document Contains Keywords*. The actual rank in the example "diet" related Websites network could look like this: Atkins Home–atkins–nutrition–carb. On the other side the (stemmed) keyword "nutrition" could have its own rank in the dimension *Keyword Appears in Documents*: nutrition–DietSite–Atkins Home.

4. User Interaction

In this example of the user interaction with the system, the course author wants to select material for the guided tour around the "diet" devoted Websites. The author enters the term "diet" into the search box and initiates processing of the results. The page of the prototype system is divided into two areas. The left side resembles the search engine result with the addition of the selection capability of the interesting documents for saving in the collection.


Save Collection Open Collection

□ Atkins Nutritionals: Home
In the News. Atkins and Folic Acid. National Folic Acid Awareness Week is an appropriate time to dispel the myth that following a low-carb **diet** means you don't ...
atkins.com/ - 34k - Cached - Similar pages

□ Public Page | Cyberdiet
Cyberdiet - Succeed at Weight Loss, ...
www.cyber**diet**.com/ - 18k - 9 Feb 2005 - Cached - Similar pages

□ **Diet** Information: **Diet** Pills, FREE Diets,Weight Loss Programs
Diet Information Free Diets - Advice About Weight Loss Programs, **Diet** & Weight Loss Pills and **Diet** Nutrition Including Questions About Low Carb Diets, Low GI ...
www.**diet**-i.com/ - 30k - Cached - Similar pages

□ Welcome to DietSite!
Here's to better Health! Finally a **diet** site that offers nutrition information like no other site. Our site covers eating plans ...
www.**diet**site.com/ - 19k - 9 Feb 2005 - Cached - Similar pages

□ The **Diet** Channel - weight loss, **diet** analysis, nutrition ...
Cutting edge nutrition information and expert **diet** advice on weight loss, sports nutrition, **diet** and cancer, heart healthy **diet**, herbs, vitamins, minerals ...
www.the**diet**channel.com/ - 22k - Cached - Similar pages

in Education

arrow keys and menus (right)
through the demo,
ely use the x,y, s, f, e, c keys
erl implementation.

Down: DIETCHAN-kw
Across: DIET-I-Kw
You Can Turn in These Directions:

Figure 5. User Interaction example in the "à la" system for education

The user can browse a network of cells in a zzstructure in the right side pane and (s)he has two dimensions to choose at a time: Across and Down. Navigation starts at the current cell, keyword "diet", which is specially marked. If a dimension which has links towards the current cell is selected, the connected cells will be shown as arrays of horizontal or vertical ranks. The user can see that this particular keyword appears on two websites: vertically "Diet Channel" and horizontally "Diet Information". Also it is immediately visible that a vertical list of keywords intersects a horizontal list of terms in one more place, keyword "weight". Therefore these two sites share two common terms. A user can then navigate the ranks up/down or left/right. Whenever a user changes the current cell, the zzstructure view might change: some new cells might be revealed, some old hidden, all depending on the current position and the two selected dimensions. When a dimension is changed, the new one will replace the old one and the view will change accordingly.

5. System Evaluation

A set of 20 teachers was selected for the evaluation. The assumption taken was that the teachers were reasonably and equally skilled in the Internet search techniques and that they are using them regularly. The users were randomly divided into two equal groups. The first group was given a task to select material for the course in their own area, using strictly a search engine and the bookmarking techniques. After a brief demonstration, the second group was instructed to perform the same task but using the "à la" tool. The groups were then switched. The duration of the sessions was limited to 1 hour. After that, they were presented with the following questionnaire for each of the systems:

Provide a grade from 1 (the worst) to 10 (the best) for each of the following questions:

- How easy was to learn to use the system?
- How friendly was the user interface?
- How effective was the system in supporting your task?
- What was the overall satisfaction with the system?

The Wilcoxon signed rank test was used to compare the obtained results, in order to show the differences between the paired observations.

Table 1. Evaluation results showing comparison to the classical solution using ranking 1 to 10

Metrics used	**Avg. rank (search engine)**	**Avg. rank ("à la" method)**	**No of <> pairs**	**Probability of identical distribution**
Method learnability	7.70	6.75	17	<= 0.06487
Friendliness of the user interface	8.00	6.70	14	<= 0.01074
Effectiveness	7.30	8.40	15	<= 0.03534
Overall user satisfaction	7.90	8.25	13	<= 0.41430

The results indicate that the initial learnability and the friendliness of the user interface are lower for the "à la" system compared to the classical solution. However, this observation is expected as the way of using the standard search engine solution is widely known. On average, the results demonstrate better effectiveness and the overall satisfaction for the "à la" system for education. On the other hand, the future work should explore the larger user population and the usage of other metrics, in order to confirm and expand the observations obtained in this trial, especially related to effectiveness which should be objectively measured.

6. Conclusions

Teachers and authors developing Web-based courseware typically face problems in locating and organising suitable learning resources. They resort to keyword-based search using searching engines and the bookmarking techniques. The "à la" (Associative Linking of Attributes) in education, presented in this paper, offers methods for improving the classical approach to the problem of authoring Web-based courses. The "à la" technique consists of enhancing the search engine based solution in the following way:

- textual documents from the search results are analysed and the two types of attributes extracted (keywords and file formats);
- relationships between attribute instances are statistically analysed and the most frequent ones established;
- attribute links are presented to a user in a browsable hypertext structure using ZigZag principles.

In order to evaluate the mentioned research ideas, the "à la" in education prototype was implemented and evaluated during a user trial. The user study looked into how easy it was to learn to use the system, how friendly the interface was, how effective was the system in supporting the user's task, and finally, what was the overall user satisfaction. The system was compared with the classical solutions of using only the search engine. A group of teachers was asked to locate and select suitable web resources for a web course. The aim of the trial was to confirm the expected solution contributions:

- Browsing the related metadata (keywords and formats) along the search results helps determining the relevancy faster by offering a sort of quick summary of the document;
- Shared keywords help establishing which documents could be semantically related;
- Extracted keywords can provide suggestions for further searching.

Results indicated that, after the initial learning effort, the "à la" prototype proved potential to have a high level of effectiveness and a better overall user satisfaction.

Using a system by a group of teachers opens up a new research direction: the possibility of utilising the system in a collaborative environment. Ideas about sharing the authoring experiences also raise personalisation issues; therefore possible future work might comprise using personalised, continuous web content mining agent.

References

[1] M. Andric, W. Hall, L. Carr, "Assisting Artifact Retrieval in Software Engineering Projects", In Proc. of the ACM Symposium on Document Engineering (DocEng), Oct. 2004, Milwaukee, Wisconsin, USA, pp. 48-50.

[2] L. Aroyo, D. Dicheva, "The New Challenges for E-learning: The Educational Semantic Web", Educational Technology & Society, Vol.7, No. 4, 2004, pp. 59-69.

[3] L. Aroyo, R. Mizoguchi, "Authoring support framework for intelligent educational systems', Intl. Conference on Artificial Intelligence in Education (AIED'03), July 2003, Sydney, Australia, pp. 362–364.

[4] L. Carr, "ZigZag for Web Browsers", resources at http://www.ecs.soton.ac.uk/~lac/zigzag, 2001.

[5] V. Devedzic, "Key Issues in Next-Generation Web-Based Education", IEEE Transactions on Systems, Man, and Cybernetics, Part C – Applications and Reviews, Vol.33, No.3, Aug. 2003, pp. 339-349.

[6] V. Devedzic, "Education and The Semantic Web", International Journal of Artificial Intelligence in Education (IJAIED), Vol.14, 2004, pp. 39-65.

[7] S. El-Beltagy, W. Hall, D. De Roure, L. Carr, "Linking in Context". In Proc. of the ACM Hypertext '01 Conf., July 2001, Arhus, Denmark, pp. 151-160.

[8] M. McGuffin, m. schraefel, "Hyperstructure: A comparison of hyperstructures: zzstructures, mSpaces, and polyarchies". In Proc. of the ACM Hypertext '04 Conf., Aug. 2004, Santa Cruz, USA, pp. 153-162.

[9] T. Nelson,"What's On My Mind". Invited talk at The 1st Wearable Computer Conf., 1998, Fairfax, USA

[10] T. Nelson, "Structure, Tradition and Possibility". In Proc. of the ACM Hypertext '03 Conf., Aug. 2003, Nottingham, UK, pp 1..

[11] S. Trausan-Matu, D. Maraschi, S. Cerri, "Ontology-Centered Personalized Presentation of Knowledge", In Proc. of the 6th International Conf. on Intelligent Tutoring Systems, ITS 2002, June 2002, Biarritz, France and San Sebastian, Spain, pp. 259-269.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Inferring learning and attitudes from a Bayesian Network of log file data

Ivon ARROYO, Beverly Park WOOLF
Department of Computer Science University of Massachusetts,
Amherst, MA. Contact: ivon@cs.umass.edu

Abstract. A student's goals and attitudes while interacting with a tutor are typically unseen and unknowable. However their outward behavior (e.g. problem-solving time, mistakes and help requests) is easily recorded and can reflect hidden affect status. This research evaluates the accuracy of a Bayesian Network to infer a student's hidden attitude toward learning, amount learned and perception of the system from log-data. The long term goal is to develop tutors that self-improve their student models and their teaching, dynamically can adapt pedagogical decisions about hints and help improve student's affective, intellectual and learning situation based on inferences about their goals and attitude.

1 Introduction

The advent of the Internet has promoted Web-based learning environments that facilitate collection of enormous student data, as a result of centralized servers and databases. Log data permit the analysis of fine-grained student actions that characterize fading of students' mistakes or the reduction of time on task [1]. The analysis of learning curves may also show how to structure and better understand the domain being taught [2]. Learning to profit from this logfile data to enhance our learning environments is one of the next greatest challenges for the AIED community.

We describe our results of creating a bayesian model from data, in which very crude and generic descriptors of students' behavior in a tutoring system are used to predict a students' goals, attitudes and learning for a large database of student actions. We present a model that shows that such dependencies do exist, describe the methodology we used to find a good model, evaluate its accuracy and identify the accuracy of alternative models. The final goal is to use the model to impact students' learning and positive attitudes towards learning, and to eventually create a module in the tutor that recomputes the model as new data arrives, thus improving it with new students' data.

This community has made recent attempts to link students' attitudes and learning to actual behavior [3, 4, 5, 6]. Aleven proposed a taxonomy of help seeking bugs and possible hints to be given by the tutoring system to encourage positive behaviors. Zhou and Conati built a Bayesian model to infer students' emotions and personality for a mathematics game. Baker observed students' behavior and classified those "gaming" the system. This paper is an integration of that past work; it merges motivation, learning, and misuse of tutoring systems in one single Bayesian model, presenting the complexity of behaviors linked to students' affect and cognition, advocating for data-driven models that integrate cognition, motivation and their expression with different behavioral patterns.

2 Data sources: Integration of survey and log files data summaries

This section describes the first step in the methodology to use observable student behavior to infer student learning and attitudes, specifically how to identify dependencies between hidden and observable variables. We used log data from Wayang Outpost, a multimedia web-based tutoring system for high school mathematics [7] to predict affective variables, e.g., the student liked the experience, was learning and was trying to challenge himself. Wayang Outpost provides step-by-step instruction to the student in the form of animations, aided with sound, which help students solve the current problem and teach concepts that are transferred to later problems. Problems were presented in a random order (no adaptive problem selection). Every interaction of student and tutor is logged in a server-side relational database, allowing researchers to record variables such as time spent, number of problems seen and speed of response. The data used in this study comes from a population of 230 15-17 year-old students from two high schools in rural and urban areas in Massachusetts. Students took a pretest and then used Wayang Outpost for about 2-3 hours. After using the tutor, students took a post-test, and answered a survey to identify their hidden attitudes and learning.

Table 1 describes the instruments used to detect students' attitudes and motivation at the end of the study, with code names for each question (in bold). In addition, we identified observable student behavior, specifically students' ways of interacting with the system, that reflect the effort or focus of attention at specific moments. They describe generic problem-solving behavior, e.g., mistakes, time, help requests and behavior in problems where the student requests help. This observable behavior falls into four categories: (1) Problem-solving behavior, e.g., average incorrect responses, specifically for those problems where help was requested; average seconds spent in any problem and where help was requested; and seconds spent between making attempts. (2) Help activity, average hints requested per problem; average hints in helped problems (when a student asks for help, how much help does she request?); average seconds spent in helped problems (time/effort the student invested when she asked for help); the percentage of helped problems in the tutoring session (how often the student asked for help). (3) Help timing, i.e. the timing of when help was sought as a percentage of all helped problems: *help before making an attempt*; *help after making an attempt*; *help after entering the correct answer*. (4) Other descriptors, past experience (correct and incorrect answers in the pre-test); gender (we had seen gender differences both in attitudes and interactions with the tutors in the past); time between pairs of attempts. The next section describes an exploratory analysis to find the

Student Perceptions of the tutor.
Learned? Do you think you learned how to tackle SAT-Math problems by using the system?
Liked? How much did you like the system?
Helpful? What did you think about the help in the system?
Return? Would you come back to the web site to use the system again if there were more problems and help for you to see? How many more times would you use it again?
Interaction with the tutor.
Audio? How much did you use the audio for the explanations?
Attitudes towards help and learning
Seriously try learn. How seriously did you try to learn from the tutoring system?
Get it over with (fast). I just wanted to get the session over with, so I went as fast as possible without paying much attention.
Challenge. I wanted to challenge myself: I wanted to see how many I could get right, asking as little help as possible.
No care help. I wanted to get the correct answer, but didn't care about the help or about learning with the software.
Help fading attitude. I wanted to ask for help when necessary, but tried to become independent of help as time went by.
Other approaches. I wanted to see other approaches to solving the problem, and thus asked for help even if I got it right.
Fear of Wrong. I didn't want to enter a wrong answer, so I asked for help before attempting an answer, even if I had a clear idea of what the answer could be.

Table 1. Post-test of student attitudes.

connection between these concrete observable variables and the more abstract and hidden ones derived from the survey.

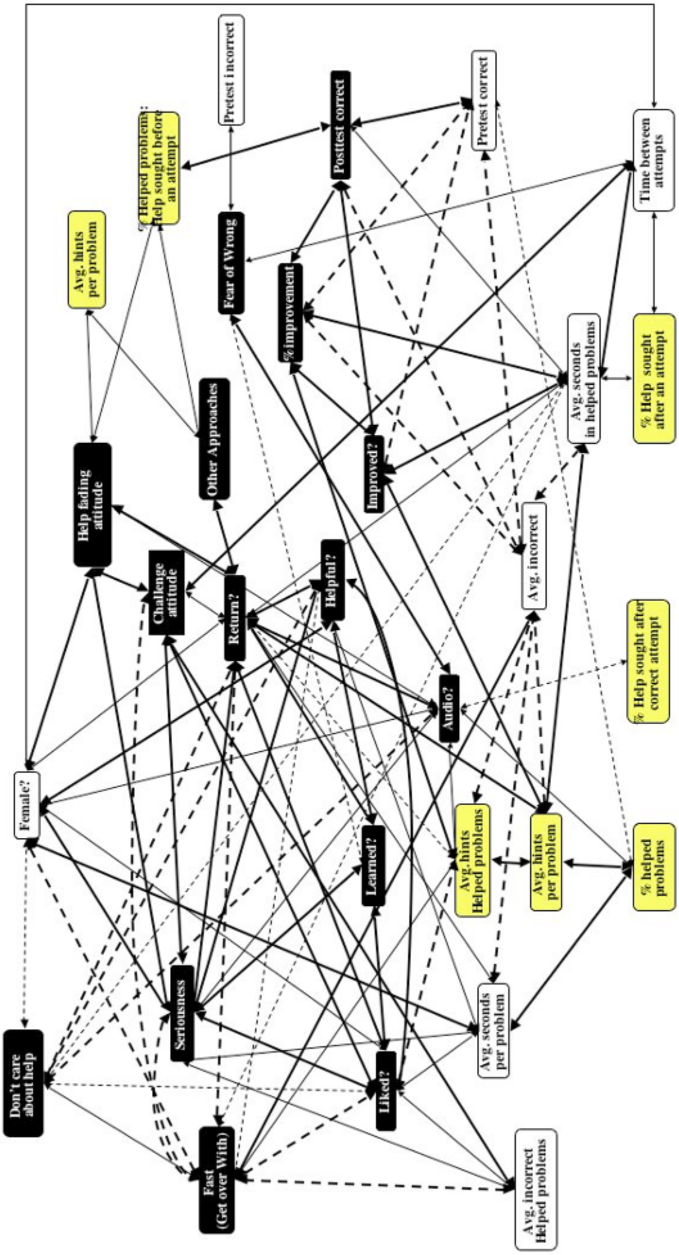


Figure 1. Correlations between hidden and observed variables. Variables that describe a student's observed interaction style (light colored nodes) are correlated with the students' hidden attitudes, feelings and learning (dark nodes) derived from the survey. Line weight indicates correlation: dashed line (- -) indicates a negative correlation; lines (—) indicate a positive correlation; thick lines indicate p<0.01 and R>0.30 - light lines indicate correlations of p<0.05

We may attempt to interpret these dependencies among variables to understand students' use of the system. For instance, learning gains from pre to post-test (%improvement) is not correlated to 'average hints seen per problem', but it is correlated to 'average hints seen in *helped* problems'. Thus, students who search deeply for help are more likely to learn. Other variables that relate to %improvement indicate that this relationship is more complex, since learning gain is not positively correlated with 'time spent in a problem,' but it is correlated to 'time spent in those problems where help was seen.' This suggests that spending much time struggling in a problem and not seeing help will not guarantee learning; instead, a student should spend significant time seeing help. Learning is inversely correlated to average incorrect attempts per problem, suggesting that students who make many incorrect responses per problem will not display a large improvement from pre to posttest. Many of these correlations are not very strong (in general, neither of them by themselves accounts for more than 15% of the variance). However, a model that integrates all these variables together should allow for a better prediction of the dependent variables that indicate success in a learning environment.

3 Identifying dependencies among variables

Bi-variate Pearson correlations were computed to search for links among the hidden and observed variables. Figure 1 shows the high number of significant correlations found among help seeking attitudes, help seeking behaviors, perceptions of the system, gender and other behaviors, such as problems seen and how often a student reported hearing the audio for explanations. Thick lines indicate a significant correlation with p<0.01 and an R>0.3, while light lines indicate significant correlations with strength p<0.05. As expected, there are dependencies among variables within a group of hidden variables, such as significant correlations among the variables that describe perceptions towards the system.

Students' general perceptions and attitudes are also correlated to many concrete behaviors in the tutor. In general, making mistakes while asking for help seems to be a positive action and is correlated to 'seriousness' and 'liking of the system,' though not directly associated to higher learning gains. It is also correlated to the 'challenge' attitude, showing that students might want to make an attempt even if they risk a wrong answer. One interesting dependency is that a high number of mistakes per problem is correlated to a higher chance of a student saying he/she wants to 'get over with' (probably just clicking through to get the answer). However, making a high number of mistakes in problems where they do request help is linked to a lower likelihood of wanting to 'get over with' the session. Interestingly, there are no strong correlations between a student's perceptions of learning and actual learning. This is consistent with past research reports that students may overestimate or underestimate their learning, and that students' perception of learning may not reflect actual learning. Interestingly, positive student attitudes are correlated with behaviors that, in turn, lead to high learning gains (e.g. 'improved?' and 'return?' are both positively correlated to 'average hints per problem'; 'Get over with' and 'Don't care about help' are negatively correlated to 'average seconds in helped problems' which is positively correlated to '% improvement' and 'post-test correct').

4 Building an integrated model of behavior, attitude and perception

The previous sections described the first step in a methodology to infer student learning gains data: A correlation was identified between hidden and observed variables. The next step is to build a complex Bayesian Network to diagnose a student's hidden variables given only observed variables. If an accurate inference of attitudes and learning can be made while the student is

'Fear of wrong'	'Challenge'	Time between attempts	Cases	Probability	
False	False	Low	43	0.64	(1)
		High	24	0.36	(2)
	True	Low	35	0.42	(3)
		High	48	0.58	(4)
True	False	Low	8	0.50	(5)
		High	8	0.50	(6)
	True	Low	7	0.32	(7)
		High	15	0.68	(8)

Table 2. Learning the conditional probability tables (CPT)
Maximum likelihood to learn conditional probability tables for 'fear of wrong' node from students' data

using the system, then the tutor can anticipate a student's posterior answers about perceptions of the system. We created a student model that is informed about past correlations results and can integrate real-time observable behavior of a student with more abstract and hidden attitudes and beliefs.

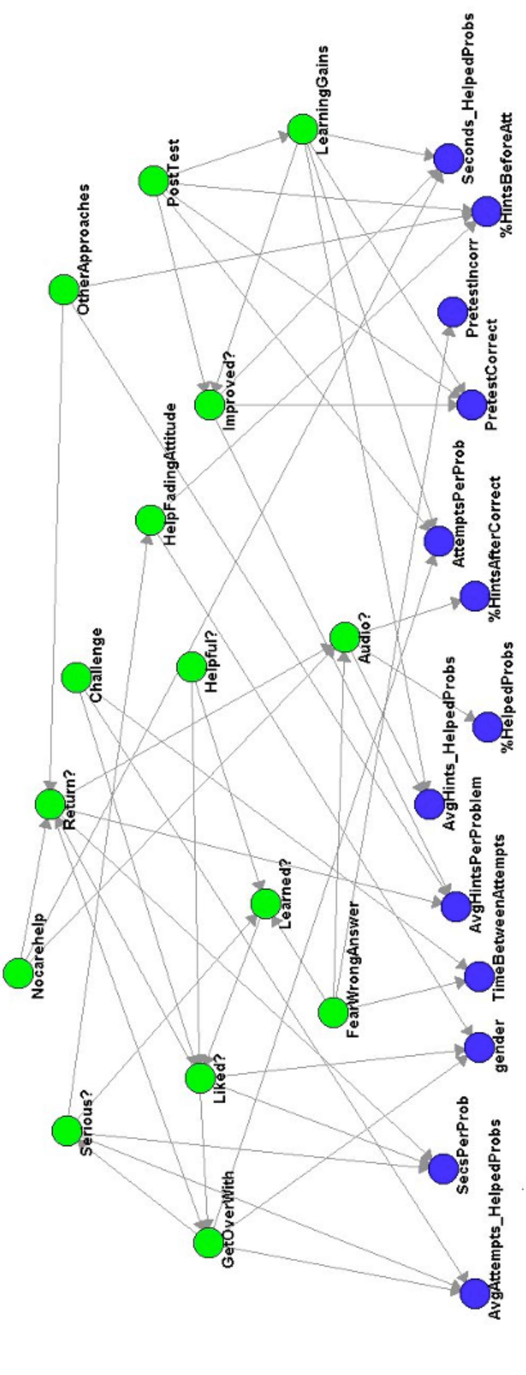


Figure 2. Structure of a Bayesian Network to infer attitudes, perceptions and learning. Nodes are the same as those in Figure 1. The bottom or leaf nodes are observable.

Bayesian networks that are learned from data can capture the complex dependencies among variables, as they to predict the probability of the truth of some unknown variables, given that a few others have been observed. We constructed the Bayesian model shown in Figure 2 that relies on the knowledge gained from correlation analyses in Figure 1, based on the fact that links in a Bayesian net express a dependency and variables that are not correlated are unlikely to be dependent on each other. A directed acyclic graph was created by: 1) eliminating the correlation links among observable variables (a naïve approach); 2) giving a single direction to the links from non-observable to observable variables (the observable variables being the leaf nodes, also known as the "outputs" or the "effects"); 3) for links between non-observable variables, creating intuitive unidirectional links (from the nodes that are more likely "causes" to the nodes that are more likely effects; 4) eliminating links that create cycles, leaving in the links that have a higher correlation strength. This resulted in a directed acyclic graph (DAG) that gave the structure of the Bayesian Network in Figure 2. Next, the parameters to the network were generated by: 1) discretizing all variables in two levels (high/low) with a median-split; 2) simplifying the model further by discarding existing links whose connecting nodes do not pass a Chi-Square test (the dependency is not maintained after making the variables discrete); 3) creating conditional probability tables (CPTs) from the cross-tabulations of the students' data ("maximum likelihood" method for parameter learning in discrete models [8]). As an example, Table 2 shows the conditional probability table attached to the node 'Time Between Attempts.' The CPT table attached to the observable node 'time between attempts' has two parents: 'fear of wrong' and 'challenge,' see Figure 1. Many interesting probabilities are captured: when a student reports a 'challenge' attitude, the chance of spending a large amount of time between subsequent attempts is higher than when a student does not report wanting to 'challenge' herself (compare (4) to (2) and (8) to (6) in Table 2). When a student reports 'fear of the wrong answer,' there is also higher likelihood of spending a long time between attempts (compare (8) to (4) and (6) to (2) in Table 2). The probability of spending a large amount of time between attempts is highest when the student reported both 'fear of wrong' and 'challenge attitude;' it is lowest when the student did not report 'fear of wrong' or did not want to 'challenge' herself.

5 Model accuracy

A 10-fold cross-validation was performed to test the accuracy of the model. The following process was repeated 25 times: the conditional probability tables were learned from 90% of students' data; the remaining 10% was used to test the model. The model was tested in the following way: the leaf nodes (observable student behavior within the tutor) were instantiated (observed) with the behavior that the student displayed (including gender and pre-test correct and incorrect). Then, the hidden nodes (attitudes, learning improvement, post-test score, perceptions of helpfulness) were inferred with the Bayesian network. If the probability of true was higher than 0.7 and the true value of the inferred node was 1 (i.e., true, or high, depending on the variable), a "hit" was produced. A hit was also produced for an inference lower than 0.3 and the actual value being a 0 (i.e., false, or low, depending the variable). A "miss" was detected when the inference was higher than 0.7 but the actual value was a 0 (or false, or low). If the inference was within the interval (0.3, 0.7), the inference was considered too uncertain and thus did not "fire." The accuracy for each node was computed as the ratio of hits to the

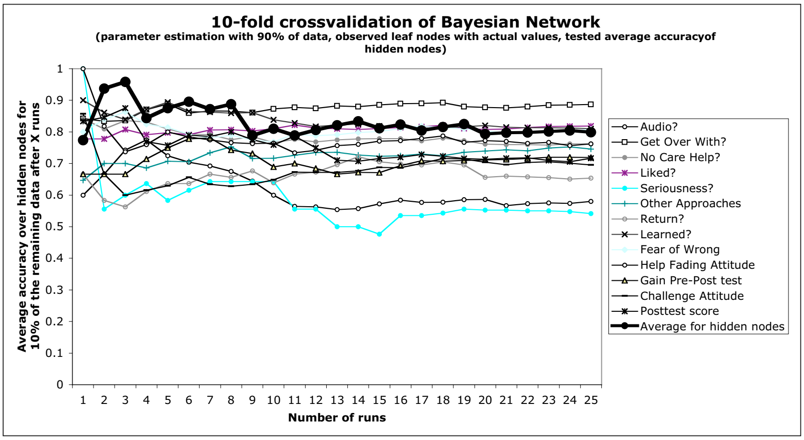


Figure 3. Accuracy of inferred nodes. This validation test measured the accuracy of the Bayesian network to learn the hidden nodes (attitudes and learning improvement) with a 10-fold crossvalidation. The graph shows the percentage of hits for all hidden nodes.

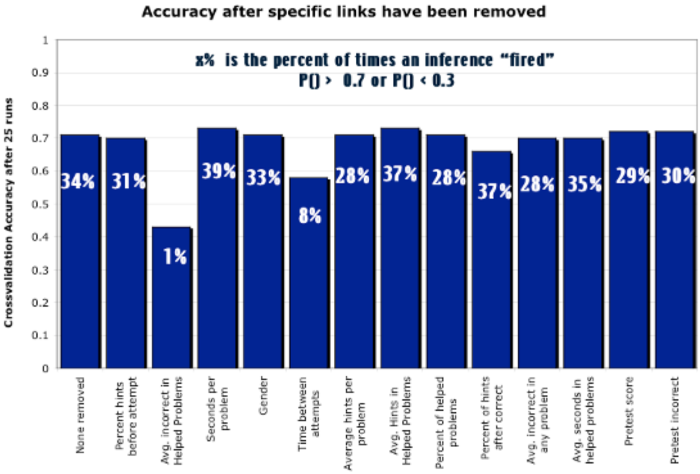


Figure 4. Removing Specific Links. Accuracy of inferences of the "Challenge Attitude" node after removing some observable-behavior nodes.

total (hits + misses). Figure 3 shows the percentage of hits for all hidden nodes, after 1 to 25 runs. Nodes with higher accuracy also contain less uncertain inferences; 90% of the 'get-over-with' inferences "fired", falling outside the (0.3,0.7) interval, while only 11% of the inferences for the 'seriousness' attitude fall outside of that interval (89% of the inferences were considered uncertain). Because only "certain" inferences were taken into account, the average trend is closer to the most accurate nodes. For some nodes, the accuracy is low, such as for students' reported 'use of audio,' or students' 'seriousness' while using the system. Seriousness relies on 'audio' for its inferences, and audio is hard to predict from observable behaviors (the audio was very embedded in the hints, so seeing/not seeing hints is not a good discriminant to determine whether they are hearing the audio). It may be hard to detect whether students are listening to the audio or not without explicitly asking about it.

6 Understanding the model

Even if models are produced from data, we think it is important to produce models that are inspectable. We may now query the model to gain knowledge and answer questions about students' learning: How does a student who demonstrates a high gain from pre-test to post-test interact with the tutor compared to one who doesn't learn? How does a motivated student behave compared to one who doesn't seem motivated? We may query the model to learn about students' learning. Table 3 shows how setting one observable-behavior node to different values, produces different inferences of '% improvement from pre-test to post-test' and for students report of 'I didn't care about help'. Students who spend higher than average seconds in a problem also have a higher chance to get higher learning gains, and also have a lower chance to report that they did not care about help.

A more detailed analysis of how behavior effects higher-level variables was carried out, by removing the links to some leaf nodes from the model and seeing how that affects the overall accuracy. Figure 4 shows that when removing links to certain observable nodes, accuracy in predicting other nodes becomes diminished. For instance, we can observe how removing the node called 'incorrect responses in helped problems' (third column from the left) affects the prediction of the 'challenge' attitude, and produces more uncertain inferences. This is important if one intends to understand which behaviors predict attitudes and learning. It may also be used to simplify the model: if an immediate child is removed but the accuracy is not affected, the link to that node can be removed, as it merely promotes over-fitting. One reason for this may be that another behavior captures the same effect. Removing links to other nodes can provide a clear sense of how certain variables affect the prediction of others and provide guidelines to improve the BBN.

Seconds in helped problems	Learning Gains. Improvement	Posterior probability	'I didn't care about help'	Posterior probability
Low	Low	0.54	True	0.27
	High	0.47	False	0.72
High	Low	0.33	True	0.08
	High	0.67	False	0.92

Table 3. A model of learning and attitude that can be inspected. This model may be queried to gain knowledge and answer questions about students' learning.

7 Summary and Future Work

We have described a methodology to build a model from log-data that integrates behavioral, cognitive and motivational variables. We showed how the methodology was applied to our bank of data for a tutoring system and how the model captures the complexity of variables that describe the student and capitalize on this dependency structure to infer the students' cognitive and affective state. We highlighted how machine learning methods and a classical statistical analysis can be combined to find an accurate model in non-exponential time. This is important when considering a large amount of behaviors and other variables, or when thinking about self-improving models that can be enhanced as new users arrive to the system. Future work relates to implementing various forms of remediation that would be triggered in certain "undesirable" situations that are linked to lower learning and negative attitudes.

8 Acknowledgements

We gratefully acknowledge support for this work from two National Science Foundation awards: 1) HRD/EHR 012080, Beal, Woolf, & Royer, "AnimalWorld: Enhancing High School Women's Mathematical Competence;" and 2) HRD/HER 0411776, Woolf, Barto, Mahdevan, Fisher & Arroyo, "Learning to Teach: the next generation of Intelligent Tutor Systems". Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the granting agencies.

9 References

[1] Corbett, A. & Anderson, J. (1992). Knowledge tracing in the ACT Programming Tutor. The Proceedings of the 14th Annual Conference of the Cognitive Science Society. Hillsdale, NJ: Lawrence Erlbaum

[2] Koedinger, K. & Santosh, M (2004). Distinguishing Qualitatively Different Kinds of Learning Using Log Files and Learning Curves. Workshop "Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes," ITS 2004.

[3] Zhou X. & Conati C. (2003). Inferring User Goals from Personality and Behavior in a Causal Model of User Affect . In Proceedings of the International Conference on Intelligent User Interfaces, pp. 211-218.

[4] Baker, R., Corbett, A.T. and Koedinger, K.R. (2001). Toward a Model of Learning Data Representations. Proceedings of the Twenty-Third Annual Conference of the Cognitive Science Society, 45-50

[5] Aleven, V., McLaren, B., Roll, I. & Koedinger, K. (2004). Toward Tutoring Help Seeking: Applying Cognitive Modeling to Meta-Cognitive Skills. In the *Proceedings of the 7th International Conference on Intelligent Tutoring Systems* (ITS-2004). Springer.

[6] de Vicente, A. & Pain, H. (2002). Informing the detection of the students' motivational state: an empirical study. In Proceedings of the 6th International Conference on Intelligent Tutoring Systems. Lecture Notes in Computer Science. Springer.

[7] Arroyo, I., Beal, C. R., Murray, T., Walles, R., Woolf, B. P. (2004b). Web-Based Intelligent Multimedia Tutoring for High Stakes Achievement Tests. *Proceedings of the 7th International Conference on Intelligent Tutoring Systems*, 468-477, Springer.

[8] Russell, S. & Norvig, P. (2002). Artificial Intelligence: A Modern Approach (2nd Edition). Chapter 14: Probabilistic Reasoning Systems.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Why Is Externally-Regulated Learning More Effective Than Self-Regulated Learning with Hypermedia?

Roger AZEVEDO, Daniel MOOS, Fielding WINTERS, Jeffrey GREENE,
Jennifer CROMLEY, Evan OLSON, and Pragati GODBOLE CHAUDHURI
University of Maryland, Department of Human Development, Cognition and Technology Lab
College Park, MD, USA, 20742 {razevedo@umd.edu}

Abstract. In this study we examined the effectiveness of self-regulated learning (SRL) and externally-regulated learning (ERL) on adolescents' learning about the circulatory system with hypermedia. A total of 128 middle-school and high school students with little knowledge of the topic were randomly assigned either to the SRL or ERL condition. Learners in the SRL condition regulated their own learning, while learners in the ERL condition had access to a human tutor who facilitated their self-regulated learning. We converged product (pretest-posttest shifts in students' mental models) with process (think-aloud) data to examine the effectiveness of self- and externally-regulated learning about a science topic during a 40-minute session. Findings revealed that the ERL condition facilitated the shift in learners' mental models significantly more than did the SRL condition. Verbal protocol data indicated that learners in the ERL condition regulated their learning by activating prior knowledge, engaging in several monitoring activities, deploying several effective strategies, and engaging in adaptive help-seeking. By contrast, learners in the SRL condition regulated their learning by using fewer monitoring activities, and using several ineffective strategies. We present design principles for adaptive hypermedia learning environments designed to foster students' self-regulated learning of complex and challenging science topics.

Introduction

Can adolescents use a hypermedia learning environment to learn about complex and challenging science topics such as the circulatory system? Learning with a hypermedia environment requires a learner to regulate his or her learning; that is, to make decisions about what to learn, how to learn it, how much time to spend on it, how to access other instructional materials, and to determine whether he or she understands the material [1,2]. Specifically, students need to analyze the learning situation, set meaningful learning goals, determine which strategies to use, assess whether the strategies are effective in meeting the learning goal(s), and evaluate their emerging understanding of the topic. They also need to monitor their understanding and modify their plans, goals, strategies, and effort in relation to contextual conditions (e.g., cognitive, motivational, and task conditions) [3,4,5]. Further, depending on the learning task, they may need to reflect on the learning session. In this study, we examine the effectiveness of self-regulated learning (SRL) and externally-regulated learning (ERL) in facilitating qualitative shifts in students' mental models (from pretest to posttest) and the use of self-regulatory processes associated with these shifts in conceptual understanding.

Contemporary cognitive and educational research has shown that the potential of hypermedia as a learning tool may be undermined by students' inability to regulate several aspects of their learning [1,6,7]. For example, students may not always deploy key metacognitive monitoring activities during learning (e.g., [8]); it has been shown that they do not engage in planning activities such as creating learning goals and activating prior knowledge (e.g., [9]); they also may predominantly use ineffective strategies such as copying information from the hypermedia environment to their notes and may navigate the hypermedia environment without any specific learning goals (e.g., [10]). One potential solution to enhancing students' regulation of their learning with hypermedia is to examine how an external (to the student's cognitive system) regulating agent, such as a human tutor may facilitate a student's self-regulated learning by prompting the student to use certain key SRL processes during learning.

We have adopted and extended Winne's [4] model of SRL by examining the role of a human tutor as an external regulating agent capable of facilitating students' self-regulated learning with hypermedia. According to his model, any scaffold (human/non-human, static/dynamic) that is designed to guide or support students' learning with hypermedia is considered a part of the task conditions. The role of scaffolds that are part of the task conditions (and therefore external to the learner's cognitive system) needs to be experimentally examined to determine their effectiveness in fostering self-regulated learning. In this study, a human tutor could potentially assist students in building their understanding of the topic by providing dynamic scaffolding during learning and facilitate students' learning by assisting them in deploying specific self-regulatory skills (e.g., activating students' prior knowledge). In so doing, a human tutor can be seen as a regulatory agent that monitors, evaluates, and provides feedback regarding a student's self-regulatory skills. This feedback may involve scaffolding students' learning by assisting them in planning their learning episode (e.g., creating sub-goals, activating prior knowledge), monitoring several activities during their learning (e.g., monitoring progress towards goals, facilitating recall of previously learned material), prompting effective strategies (e.g., hypothesizing, drawing, constructing their own representations of the topic), and facilitating the handling of task demands and difficulty. Empirically testing the effectiveness of self-regulated learning and externally-regulated learning can elucidate how these different scaffolding methods facilitate students' self-regulated learning and provide evidence that can be used to inform the design of hypermedia learning environments. In this paper we focus on two research questions— 1) *Do different scaffolding conditions influence learners' ability to shift to more sophisticated mental models of the circulatory system?* 2) *How do different scaffolding conditions influence learners' ability to regulate their learning?*

1. Method

1.1 Participants. Participants were 128 high school and middle school students from local schools in a large mid-Atlantic city in the United States of America. The mean age of the 67 high school students was 15 years and the mean age of the 61 middle school students was 12 years.

1.2 Paper-and-Pencil Measures. The paper-and-pencil materials consisted of a consent form, a participant questionnaire, a pretest, and a posttest. All of the paper-and-pencil materials were constructed in consultation with a nurse practitioner who is a faculty member at a school of nursing in a large mid-Atlantic university and a science teacher. The pretest consisted of a sheet which contained the instruction, *"Please write down everything you can about the circulatory system. Be sure to include all the parts and their purpose, explain how they work both individually and together, and also explain how they contribute to the healthy functioning of the body"* (mental model essay). The pretest and posttest were identical.

1.3 Hypermedia Learning Environment (HLE). During the training phase, learners were shown the contents and features of the circulatory system, blood, and heart articles in the hypermedia environment. Each of these relevant articles contained multiple representations of information—text, static diagrams, and a digitized animation depicting the structure, behavior, and functioning of the circulatory system. Together these three articles comprised 16,900 words, 18 sections, 107 hyperlinks, and 35 illustrations. During the experimental phase, the learners used the hypermedia environment to learn about the circulatory system. Learners were allowed to use all of the system features including the search functions, hyperlinks, table of contents, multiple representations of information, and were allowed to navigate freely within the environment.

1.4 Procedure. The first five authors tested participants individually in all conditions but did not tutor the students. The third author acted as the tutor in the ERL condition. Learners were randomly assigned to one of two conditions: SRL (n = 65) and ERL (n = 63). The learners were given 20 minutes to complete the pretest (mental model) essay. Then, the experimenter provided instructions for the learning task. The following instructions were read and presented to the participants in writing.

Self-Regulated Learning (SRL) Condition. For the SRL condition, the instructions were: "*You are being presented with a hypermedia learning environment, which contains textual information, static diagrams, and a digitized video clip of the circulatory system. We are trying to learn more about how students use hypermedia environments to learn about the circulatory system. Your task is to learn all you can about the circulatory system in 40 minutes. Make sure you learn about the different parts and their purpose, how they work both individually and together, and how they support the human body. We ask you to 'think aloud' continuously while you use the hypermedia environment to learn about the circulatory system. I'll be here in case anything goes wrong with the computer or the equipment. Please remember that it is very important to say everything that you are thinking while you are working on this task.*"

Externally-Regulated Learning (ERL) Condition. The instructions for the ERL condition were identical to those for the SRL condition. In addition, learners had access to a human tutor who was trained to facilitate students' self-regulated learning (SRL) by:
(1) prompting students to activate their prior knowledge (PKA);
(2) prompting several monitoring activities by having students compare what they were learning with previously learned material (FOK), monitor their emerging understanding during the task (JOL), and monitor their progress towards their goals (MPTG); and,
(3) prompting students to use several effective strategies to learn, such as hypothesizing, coordinating informational sources, drawing, mnemonics, inferences and summarization, and while meeting the same overall learning goal as the participants in the SRL condition. The human tutor was instructed not to provide additional content knowledge not included in the sections the students used in the hypermedia environment during the learning episode. This macro-script was modified from tutoring scripts found in the literature (e.g., [11,12]) and current empirical findings on SRL and hypermedia (e.g., [8,9,10]). The tutor used the following script to assist the learner in regulating his/her learning:

(1) Ask student what he/she already knows about the circulatory system, set some goals, and determine how much time to spend on each goal.
(2) Start by having student read introduction section of the *circulatory system* article: Prompt student to summarize; learn about blood flow through the heart by using several strategies (e.g., coordinating informational sources); ask several questions to determine students' understanding of the various issues related to flow; make sure student understands the purpose of lungs; suggest watching the animation to integrate all the information; assess whether student has good understanding (i.e., can he/she explain the entire process in his/her own words). If not, then have student draw and label a diagram of the heart and assess their understanding [repeat (2)]. If yes, then proceed to the *blood vessel diagram*.
(3) Revisit global learning goal, give time reminder, state which goals have been met and which still need to be satisfied.
(4) Have student read text for the *blood vessels diagram*. Assess student's understanding. If the student did not understand, then have him/her re-read the introduction, major components, and diagrams comparing veins and arteries, and then assess understanding [repeat (4)]. If the student demonstrates that he/she understood, then proceed to the *blood article*.
(5) Revisit global learning goal, give time reminder, state which goals have been met and which still need to be satisfied.
(6) Have students read the *blood article.* If the student did not understand, then have him/her re-read the introduction and role of blood section, and then assess understanding [repeat (6)]. If the student demonstrates that he/she understood, then review their notes and drawings.

(7) Revisit global learning goal, give time reminder, state which goals have been met and which still need to be satisfied.

In both conditions, an experimenter remained nearby to remind participants to keep verbalizing when they were silent for more then three seconds (e.g., *"Say what you are thinking"*). All participants were reminded of the global learning goal (*"Make sure you learn about the different parts and their purpose, how they work both individually and together, and how they support the human body"*) as part of their instructions for learning about the circulatory system. All participants had access to the instructions (which included the learning goal) during the learning session. Participants in the ERL condition also had access to the tutor. All participants were given 40 minutes to use the hypermedia environment to learn about the circulatory system. Participants were allowed to take notes and draw during the learning session, although not all chose to do so. The posttest was administered immediately following the learning session, and all participants independently completed the posttest in 20 minutes without their notes or any other instructional materials by writing their answers on the sheet provided by one of the experimenters.

1.5 Coding and Scoring. In this section we describe the coding of the students' mental models, the segmentation of the students' verbalizations while they were learning about the circulatory system, the coding scheme we used to analyze the students' regulatory behavior, and inter-rater agreement.

Mental models. Our analyses focused on the shifts in participants' mental models based on the different scaffolding conditions. We followed Azevedo and colleagues' method [8,9,10] for analyzing the participants' mental models, which is based on Chi and colleagues' research [13,14,15]. A student's initial mental model of how the circulatory system works was derived from their statements on the pretest essay. Similarly, a student's final mental model of how the circulatory system works was derived from their statements from the essay section of the posttest. Our scheme consists of 12 mental models which represent the progression from no understanding to the most accurate understanding: (a) no understanding, (b) basic global concept, (c) basic global concept with purpose, (d) basic single loop model, (e) single loop with purpose, (f) advanced single loop model, (g) single loop model with lungs, (h) advanced single loop model with lungs, (i) double loop concept, (j) basic double loop model, (k) detailed double loop model, and (l) advanced double loop model. See [8, p. 534-535] for a complete description of the necessary features for each of the 12 mental models.

The third and fifth author scored the students' pretest and posttest mental models by assigning the numerical value associated with the mental models described in [8, p. 534-535]. The values for each student's pretest and posttest mental model were recorded and used in a subsequent analysis to determine the qualitative shift in their conceptual understanding based on their pretest and posttest mental models (see inter-rater agreement below).

Learners' verbalizations and regulatory behavior. The raw data collected from this study consisted of 5,120 minutes (85.3 hours) of audio and video tape recordings from 128 participants, who gave extensive verbalizations while they learned about the circulatory system. During the first phase of data analysis, a graduate student transcribed the think-aloud protocols from the audio tapes and created a text file for each participant. This phase of the data analysis yielded a corpus of 1,823 single-spaced pages (M = 14.24 pages/participant) with a total of 551,617 words (M = 4,309.51 words/participant). These data were used to code the learners' SRL behavior.

Our model of SRL was used to analyze the learners' regulatory behavior [see 8,9,10]. It is based on several current models of SRL [3,4,5]. It includes key elements of these models (i.e., Winne's [4] and Pintrich's [3] formulation of self-regulation as a four-phase process), and extends these key elements to capture the major phases of self-regulation. These are: (a) planning and goal setting, activation of perceptions and knowledge of the task and context, and the self in relationship to the task; (b) monitoring processes that represent metacognitive awareness of different aspects of the self, task, and context; (c) efforts to control and regulate different aspects of the self, task, and context; and, (d) various kinds of reactions and reflections on the self and the task and/or context. Azevedo and colleagues' model also includes SRL variables derived from students' self-regulatory behavior that are specific to learning with a hypermedia environment (e.g., coordinating informational sources). Due to space limitations, this paper focuses solely on the students' SRL behavior.

The descriptions and examples from the think-aloud protocols of the planning, monitoring, strategy use, and task difficulty and demands variables used for coding the learners' regulatory behavior are presented in Azevedo and Cromley [8, p. 533-534]. We used Azevedo and colleagues' SRL model to re-segment the data from the previous data analysis phase. This phase of the data analysis yielded 19,870 segments (M = 155.23/participant) with corresponding SRL variables. The fifth author was trained to use the coding scheme and coded all of the transcriptions by assigning each coded segment with one of the SRL variables.

Inter-rater agreement. Inter-rater agreement was established by training the third and fifth authors to use the description of the mental models developed by Azevedo and colleagues [8,9,10]. They independently coded all selected protocols (pre- and posttest essays of the circulatory system from each participant). There was agreement on 246 out of a total of 256 student descriptions, yielding an inter-rater agreement of .96. Inter-rater agreement was also established for the coding of the learners' regulatory behavior by comparing the individual coding of several authors with that of the fifth author. The second author independently re-coded 15,276 protocol segments (77%). There was agreement on 15,123 out of 15,276 segments yielding an inter-rater agreement of .98. Inconsistencies were resolved through discussion between the two raters.

2. Results

2.1 Question 1: Do different scaffolding conditions influence learners' ability to shift to more sophisticated mental models of the circulatory system? Due to the qualitative nature of the mental models used to measure learners' understanding of the circulatory system (from pretest to posttest), we conducted a chi-square analysis to determine whether there was a significant difference in the number of learners, across conditions, whose conceptual understanding did not shift (i.e., pretest and posttest mental models were identical), or those whose mental model shifted from a low level of understanding to an intermediate level of understanding (i.e., from pretest mental model of 1 through 6 to posttest mental model of 7 or 8), or those who went from an intermediate level of understanding to a high level of understanding (i.e., from pretest mental model of 7 or 8 to posttest mental model of 9 through 12), or those whose mental model shifted from a low level of understanding to a high level of understanding (i.e., from pretest mental model of 1 through 6 to posttest mental model of 9 through 12).

A 4 X 2 (mental model shift by scaffolding condition) chi-square test revealed a significant difference in the frequency distribution of learners' mental model shifts by scaffolding condition (χ^2 [3, N = 128] = 7.976, p = .05). Overall, the ERL condition led to a significantly higher number of learners shifting to more sophisticated mental models (ERL = 49%, SRL = 31%). The ERL condition led to the highest frequency of learners shifting from a low level of understanding to a high level of understanding (ERL = 25%, SRL = 11%), and the highest frequency of learners shifting from an intermediate level of understanding to a high level of understanding (ERL = 17%, SRL = 9%). In contrast, the SRL condition led to the highest frequency of learners shifting from a low level of understanding to an intermediate level of understanding (SRL= 11%, ERL = 6%).

2.2 Question 2: How do different scaffolding conditions influence learners' ability to regulate their learning? In this section we present the results of a series of chi-square analyses that were performed to determine whether there were significant differences in the distribution of middle school and high school learners' use of SRL variables across the two conditions. We examined how participants regulated their learning of the circulatory system by calculating how often they used each of the variables related to the four main SRL categories of *planning, monitoring, strategy use,* and *handling task difficult and demands*. The number of learners using each SRL variable above the median proportion across conditions and the results of the chi-square tests are presented in Table 1.

Planning. Chi-square analyses revealed significant differences in the number of learners who used two of the four planning variables above the median proportion across the two conditions. Overall, a significantly larger number of learners in the ERL condition planned their learning by *activating their prior knowledge* and *planning* (see Table 1).

Monitoring. Chi-square analyses revealed significant differences in the number of learners who used five of the seven variables related to monitoring above the median proportion across the two conditions. Learners in the ERL condition monitored their learning by using *feeling of knowing* (FOK), *judgment of learning* (JOL), and *monitoring their progress toward goals*. In contrast, learners in the SRL condition monitored their learning mainly by *evaluating the content* of the hypermedia environment and *self-questioning* (see Table 1).

Strategies. Chi-square analyses revealed significant differences in the number of learners who used 12 of the 16 strategies above the median proportion across the two conditions. A significantly larger number of learners in the ERL condition used *hypothesizing, coordinating of information sources, drawing*, *using mnemonics, using inferences,* and *summarizing* to learn about the circulatory system. In contrast, a larger number of learners in the SRL condition learned by engaging in *free searching*, *goal-directed searching*, *selecting a new informational source, re-reading, memorization,* and *taking notes* (see Table 1).

Task difficulty and demands. Chi-square analyses revealed significant differences in the number of learners who used three of the five SRL variables related to task difficulty and demands above the median proportion across the two conditions. A significantly greater number of learners in the ERL condition handled task difficulties by *seeking help* from the tutor. In contrast, a significant number of learners in the SRL condition dealt with task difficulty and demands by *controlling the context* and *time and effort planning* (see Table 1).

3. Implications of Fostering Self-Regulated Learning with Hypermedia

Our results show that students experience certain difficulties when regulating their own learning of a complex science topic with hypermedia. By contrast, externally-regulated learning provided by a human tutor significantly relates to a higher proportion of students' experiencing qualitative shifts in their mental models of such complex topics. Our findings can inform the design of specific SRL variables to foster students' self-regulated learning with hypermedia. Based on the four SRL categories of *planning, monitoring, strategy usage,* and *task difficulties and demands*, we propose design guidelines for how specific SRL variables can be addressed to foster students' self-regulated learning with hypermedia.

Within the category of *planning*, our results suggest that prior knowledge activation and planning are key SRL variables for a hypermedia environment to scaffold. To foster prior knowledge activation, prior to commencing with the learning task, the student could be asked to recall everything they can about the topic being learned, and they could view annotations of the nodes already navigated [16]. Students could also be instructed to plan their learning within a hypermedia environment by requiring them to set goals for the learning session or have them select from a list of sub-goals presented by the environment.

Table 1. Proportion of Adolescents' Using Self-Regulated Learning Variables Above the Median Proportion, by Condition.

Variable	Self-Regulated Learning (SRL) (n = 65)	Externally-Regulated Learning (ERL) (n = 63)	χ^2	*p*
Planning				
Prior Knowledge Activation	7(11%)	**56(89%)**[b]	78.114	0.000
Planning	8(12%)	17(27%)[b]	4.385	0.036
Sub-Goals	**36(55%)**	28(44%)	1.532	0.216
Recycle Goal in Working Memory	14(22%)	14(22%)	.009	0.925
Monitoring				
Content Evaluation	**47(72%)**[a]	17(27%)	26.288	0.000
Self-Questioning	21(32%)[a]	9(14%)	5.791	0.016
Judgment of Learning (JOL)	13(20%)	**51(81%)**[b]	47.543	0.000
Feeling of Knowing (FOK)	16(25%)	**48(76%)**[b]	34.040	0.000
Monitoring Progress Toward Goals	21(32%)	**41(65%)**[b]	13.757	0.000
Monitor Use of Strategies	11(17%)	13(21%)	0.289	0.591
Identify Adequacy of Information	31(48%)	**33(52%)**	0.281	0.596
Strategy Use				
Selecting New Informational Source	**47(72%)**[a]	17(27%)	26.288	0.000
Re-Reading	**43(66%)**[a]	21(33%)	13.785	0.000
Goal-Directed Search	16(25%)[a]	4(6%)	8.097	0.004
Free Search	17(26%)[a]	5(8%)	7.459	0.006
Memorization	15(23%)[a]	5(8%)	5.563	0.018
Taking Notes	**39(60%)**[a]	25(40%)	5.283	0.022
Hypothesizing	5(8%)	**40(63%)**[b]	43.696	0.000
Coordinating Informational Sources	18(28%)	**46(73%)**[b]	26.288	0.000
Draw	18(28%)	**46(73%)**[b]	26.288	0.000
Mnemonics	1(2%)	12(19%)[b]	10.749	0.001
Inferences	24(37%)	**40(63%)**[b]	9.033	0.003
Summarization	26(40%)	**38(60%)**[b]	5.283	0.022
Read Notes	14(22%)	23(37%)	3.488	0.062
Read New Paragraph	6(9%)	4(6%)	0.369	0.544
Find Location in Environment	17(26%)	19(30%)	0.254	0.614
Knowledge Elaboration	23(35%)	22(35%)	0.003	0.956
Task Difficulty and Demands				
Control of Context	**56(86%)**[a]	8(13%)	69.048	0.000
Time and Effort Planning	26(40%)[a]	6(10%)	15.848	0.000
Help Seeking Behavior	13(20%)	**51(81%)**[b]	47.543	0.000
Expect Adequacy of Information	31(48%)	25(40%)	0.834	0.361
Task Difficulty	26(40%)	16(25%)	3.095	0.079

Note: Degrees of freedom = 1 and n = 128 for all analyses.
Note. The **bold** type indicates the variable was used above the median frequency by more than 50% of learners.
[a] SRL group made the greatest contribution to chi-square for this variable.
[b] ERL group made the greatest contribution to chi-square for this variable.

Our results indicate that several *monitoring activities* such as feeling of knowing (FOK), judgment of learning (JOL), and monitoring progress towards goals are particularly crucial to learning. To foster judgment of learning, a prompt could be made to have the students periodically rate their understanding on a Likert-type scale. A planning net could be presented at different intervals throughout the learning to aid in off-loading for monitoring progress toward goals.

There are numerous effective *strategies* that could be scaffolded in a hypermedia environment, including hypothesizing, coordinating informational sources, drawing, mnemonics, making inferences, and summarization. A major challenge with hypermedia is its inability to detect, trace, and model effective strategies and ineffective strategies [17]. Prompts and feedback could be designed to encourage effective strategies and discourage students from using ineffective strategies. For example, mnemonics scaffolding can be provided as appropriate, and drawing could be fostered via prompting when a diagram and text with relevant information are being viewed by the learner. By adding a drawing tool, a student could construct and externalize their current understanding of some aspect of the topic.

Within the category of *task difficulty and demands*, help-seeking is clearly linked to higher learning outcomes and should be scaffolded within a hypermedia environment. One challenge is to design an environment that can provide help for different aspects of the learning task. For example, a student could select the following (from a long list of items phrased as sentences) from a HELP feature—whether the current content is relevant for the current goal, get an explanation of some complex biological mechanism, determine how to coordinate multiple informational sources, etc. To close, our findings have lead us to some suggestions for how processes activated in self-regulated learners can be implemented in hypermedia environments so that these environments can foster students' self-regulated learning and conceptual understanding of complex science topics [1,6,16,17,18].

4. Acknowledgements

This research was supported by funding from the National Science Foundation (REC#0133346) awarded to the first author. The authors would like the thank Megan Clark, Jessica Vick for assistance with data collection, and Angie Lucier, Ingrid Ulander, Jonny Meritt, and Neil Hofman for transcribing the audio data.

References

[1] Azevedo, R. (in press). The role of self-regulated learning in using technology-based environments as metacognitive tools to enhance learning. *Educational Psychologist*

[2] Lajoie, S.P., & Azevedo, R. (in press). Teaching and learning in technology-rich environments. In P. Alexander & P. Winne (Eds.), *Handbook of educational psychology* (2nd ed.). Mahwah, NJ: Erlbaum.

[3] Pintrich, P.R. (2000). The role of goal orientation in self-regulated learning. In M. Boekaerts, P. Pintrich, & M. Zeidner (Eds.), *Handbook of self-regulation* (pp. 451-502). San Diego, CA: Academic Press.

[4] Winne, P.H. (2001). Self-regulated learning viewed from models of information processing. In B. Zimmerman & D. Schunk (Eds.), *Self-regulated learning and academic achievement: Theoretical perspectives* (pp. 153-189). Mahwah, NJ: Erlbaum.

[5] Zimmerman, B. (2000). Attaining self-regulation: A social cognitive perspective. In M. Boekaerts, P. Pintrich, & M. Zeidner (Eds.), *Handbook of self-regulation* (pp. 13-39). San Diego, CA: Academic Press.

[6] Jacobson, M. (in press). *From non-adaptive to adaptive educational hypermedia: Theory, research, and design issues*.

[7] Shapiro, A., & Niederhauser, D. (2004). Learning from hypertext: Research issues and findings. In D. H. Jonassen (Ed.). *Handbook of Research for Education Communications and Technology (2nd ed).* Mahwah, NJ: Lawrence Erlbaum.

[8] Azevedo, R., & Cromley, J.G. (2004). Does training on self-regulated learning facilitate students' learning with hypermedia? *Journal of Educational Psychology, 96*(3), 523-535.

[9] Azevedo, R., Guthrie, J.T., & Seibert, D. (2004). The role of self-regulated learning in fostering students' conceptual understanding of complex systems with hypermedia. *Journal of Educational Computing Research, 30*(1), 87-111.

[10] Azevedo, R., Cromley, J.G., & Seibert, D. (2004). Does adaptive scaffolding facilitate students' ability to regulate their learning with hypermedia? *Contemporary Educational Psychology, 29*, 344-370.

[11] Chi, M.T.H. (1996). Constructing self-explanations and scaffolded explanations in tutoring. *Applied Cognitive Pscyhology, 10*, S33-S49.

[12] Graesser, A.C., Person, N.K., & Magliano, J.P. (1995). Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied Cognitive Pscyhology, 9*, 495-522.

[13] Chi, M.T.H., de Leeuw, N., Chiu, M.-H., & LaVancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science, 18*, 439-477.

[14] Chi, M.T.H., Siler, S., & Jeong, H. (2004). Can tutors monitor students' understanding accurately? *Cognition and Instruction, 22*, 363-387.

[15] Chi, M.T.H., Siler, S., Jeong, H., Yamauchi, T., & Hausmann, R. (2001). Learning from human tutoring. *Cognitive Science, 25*, 471-534.

[16] Brusilovsky, P. (2001). Adaptive hypermedia. *User Modeling and User-Adapted Interaction, 11*, 87-110.

[17] Brusilovsky, P. (2004). Adaptive navigation support in educational hypermedia: The role of student knowledge level and the case for meta-adaptation. *British Journal of Educational Technology, 34*(4), 487-497.

[18] Azevedo, R. (2002). Beyond intelligent tutoring systems: Computers as MetaCognitive tools to enhance learning? *Instructional Science, 30*(1), 31-45.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Motivating Appropriate Challenges in a Reciprocal Tutoring System

Ari BADER-NATAL [1], Jordan POLLACK
DEMO Lab, Brandeis University

Abstract. Formalizing a student model for an educational system requires an engineering effort that is highly domain-specific. This model-specificity limits the ability to scale a tutoring system across content domains. In this work we offer an alternative, in which the task of student modeling is not performed by the system designers. We achieve this by using a reciprocal tutoring system in which peer-tutors are implicitly tasked with student modeling. Students are motivated, using the *Teacher's Dilemma*, to use these models to provide appropriately-difficult challenges. We implement this as a basic literacy game in a spelling-bee format, in which players choose words for each other to spell across the internet. We find that students are responsive to the game's motivational structure, and we examine the affect on participants' spelling accuracy, challenge difficulty, and tutoring skill.

Keywords. Teacher's Dilemma, reciprocal tutoring system

1. Introduction

Reciprocal tutoring systems offer an interactive environment for learning [2,3]. Chan and Chou define reciprocal tutoring as "a protocol of learning activity, where two or three agents (an agent is a computer or a student) take turns to play the roles of a 'tutor' and a 'tutee'" [2]. One reason that these systems are of interest is that they can potentially avoid the complex engineering effort required to formalize domain-specific student models. This can be avoided by transferring the responsibility of model-building to the peer helper, using human-in-the-loop techniques, similar to Kumar, et al. [7]. In order to realize this, however, we must motivate peers to appropriately challenge one another. This is a problem, as there is often a *motivation gap* between an activity's educational objectives and its motivational meta-structure. Such gaps are now beginning to be identified. Magnussen and Misfeldt reported on the behavior that they observed when students began using their educational multi-player game, in which players learned how to excel at the game while avoiding the educational challenges involved [8]. Baker, et al. also identified intentional subversion of tutoring systems as an observed problem [1]. In this paper, we seek to recognize and attempt to close these motivation gaps.

We present the foundation upon which this alternative can be based – the *Teacher's Dilemma* (TD). With participants taking on the task of student modelling, the tu-

[1] Corresponding Author: Ari Bader-Natal. Brandeis University, Computer Science Department MS018. Waltham MA 02454. USA. Tel.: +1 781 736 3366; Fax: +1 781 736 2741; E-mail: ari@cs.brandeis.edu.

TD_T	*Easy Task*	(E)	*Hard Task*	(H)
$+$	verification	v	joy	j
$-$	remediation	r	confirmation	c

TD_S	*Task*	
$+$	pass	p
$-$	fail	f

Figure 1. The *Teacher's Dilemma*: TD Teacher-Matrix on left, TD Student-Matrix on right. $+$ and $-$ denote correct and incorrect Student responses.

T_{points}	*Task of difficulty* (d)
$+$	$dv + (1 - d)j$
$-$	$dr + (1 - d)c$

S_{points}	
$+$	p
$-$	f

Figure 2. The number of Teacher-Points awarded (for challenge selection) is determined by its difficulty and the accuracy of the response. The four variables v, j, r, c are defined in the game's TD matrix (Figure 1). The number of Student-Points awarded is dependent only on Student response correctness.

toring system must provide only tutor motivation and interaction facilitation. This has been implemented as a web-based system, *Spellbee*, that was designed from the ground-up to explore these ideas. It has been publicly available for over a year, at http://www.spellbee.org. In this paper, we first examine the validity of our assumption that a player's challenge-selection strategy is influenced by the underlying motivational structure of the TD, and then examine change in player behavior over time with respect to spelling accuracy, word difficulty, typing speed, and tutoring skill.

2. Foundation: The Teacher's Dilemma

The Teacher's Dilemma presented here originates from Pollack and Blair's formulation of the *Meta-Game of Learning* [9], and has more recently been pursued by Sklar and colleagues [4,10]. The intuition behind the TD is that providing students with excessively difficult or excessively easy challenges is counter-productive, while providing appropriately challenging tasks is more valuable. The four educational extremes defining the TD are *verification* of student success at easy tasks, *joy* of student success at difficult tasks, *remediation* of student failure at easy tasks, and *confirmation* of student failure at difficult tasks. The TD provides a simple framework for describing various combinations of these educational goals. Using the TD, a teaching strategy can be described by the values a teacher attributes to each of these goals. See Figure 1.

The application of the TD to reciprocal tutoring is done by transforming the TD's representation of teaching strategy from a *model* to a *game-theoretic* formulation. Strategies in this game correspond to selecting challenges of varying levels of difficulty. The payoff values for these strategies are based on the adopted valuations (from the TD Teacher-Matrix), the level of difficulty of the challenge selected, and the accuracy of the other player's response. Figure 2 details how these payoffs are calculated for players.

The novel value of this meta-game is that players who may have no tutoring experience are effectively learning to provide the same sorts of challenges as those provided by a "model" teacher (as exemplified by the TD matrix chosen.) Improving at the TD meta-game corresponds to more closely emulating this model teacher. Given an appropriate TD Teacher-Model, pairs of students could be organized to act as tutors for one another, providing each other with increasingly appropriate challenges. Using this model, we create an entire learning community based upon participants interacting in this manner.

3. Implementation: Spellbee

In order to further explore the ideas presented above, we have built a reciprocal tutoring network for the educational domain of spelling that is based on the Teacher's Dilemma. This system, *Spellbee*, was designed for use by students in grades 3-7, and takes the form of an online educational activity[1]. Spellbee.org has been actively used for a year, during which time over 4,500 people have participated, including approximately 100 teachers and over 1,300 students of those teachers[2]. In this section, we discuss the motivational structure of the game, the mechanics of game play, and metrics for assessing challenge difficulty in this section.

3.1. Motivational Structure

The underlying motivational structure of Spellbee is derived directly from the formulation of the TD, and is presented in Figures 1 and 2. In Spellbee, each player alternates between their roles as *problem-selector* (TD's Teacher-Role) and *problem-solver* (TD's Student-Role.) When attempting to spell a word, players receive points according to a Student-Matrix in which $p = 10$ and $f = 0$ (Correct spelling is rewarded, and incorrect spelling is not.) When selecting a word for a partner, players are presented with all word-choices and corresponding $+$ and $-$ row calculated from the TD's Teacher-Matrix, given the difficulty of the word. We set the parameters of the Teacher-Matrix to $v = 0$, $j = 10$, $r = 10$, $c = 0$, in order to reward students for probing both the strengths and the weaknesses of their partner's abilities. This matrix was designed to motivate players to seek out both the hardest words that their partner might be able to correctly spell and the easiest words that their partner might not yet know.

The game itself is competitive in the sense that the partner that accrues more points (sum of Student- and Teacher-Points) wins the game. A few publicly-displayed high-score lists are maintained on the website, providing players with additional motivation to take the game-points seriously. In Section 4, we will examine the degree to which players are aware of and sensitive to the underlying motivational structure.

3.2. Game-Play

A student accesses Spellbee online at http://www.spellbee.org, and uses their pseudonym to log in. Upon entering the system, a student is placed in a *playground* of currently-available players. Mutual interest between a pair of players triggers the beginning of a new game. A game consists of a sequence of seven rounds. In each round, a player first selects a word (from a list of seven options) for their partner to attempt to spell. Each word is accompanied by a pair of point-values, determined by Figure 2. Game-play is symmetric, so both partners are concurrently selecting words. After both players select words, the word-challenges are exchanged and each attempts to spell the word that the other provided. The word-challenges are presented in a multi-modal fashion: A

[1] Spelling was selected because we recognized that for classroom adoption, our game content must coincide significantly with curricular content.

[2] This counts unique registered players who played and completed games. Teachers are self-identified upon registration, and are only counted here if they subsequently register some number of students, and those students later play and complete games.

sentence that contains the word is displayed visually, with the word-challenge blanked-out, and that sentence is also presented audibly, by playing a pre-recorded audio clip of the sentence[3]. A player is given a limited amount of time to spell the word. After spelling the word, the student first gets feedback on the accuracy of their own attempt, and then gets feedback on the accuracy of their partner's attempt. This concludes the round, and the next round begins.

3.3. Word-Difficulty Metric

In order to apply the Teacher's Dilemma to reciprocal tutoring, some measure of a challenge's level of difficulty must be available. This metric might be defined *a priori*, might be estimated using some heuristic, or might be empirically-based. In the spelling domain, we initially started with a rough heuristic[4], but quickly switched to a metric based on a particularly well-suited empirical data-set. Greene's *The New Iowa Spelling Scale* [6] aggregates data from approximately 230,000 students across the United States in grades 2-8 attempting to spell words drawn from a list of over 5,000 frequently-used words. For each word, the study calculates the percentage of students of each grade-level that correctly spelled the word. Despite being dated, this data was ideal for our needs, and so we used these grade-specific percentages as our measure of word-challenge difficulty.

4. Experiment: On Motivation

An important assumption underlying claims of Spellbee's adaptability is that players are sensitive to changes in the TD Teacher-Matrix used in the game, and this matrix can influence a player's challenge-selection strategy. We examine the validity of this assumption in a set of classroom-based experiments.

The following was done using an early Spellbee prototype in a controlled classroom setting. Students were divided into four randomly-assigned groups, and were restricted to playing games with others in the same group. Each group played using a unique TD Teacher-Matrix, as specified in Figure 3. Students in group G_1 are rewarded most for asking easy questions, independent of their partner's success or failure at responding (*Reward Easy* game). Students in G_2 are rewarded most for either asking difficult questions that their partner can correctly answer *or* easy questions that their partner cannot answer correctly (*Teacher's Dilemma* game). Students in G_3 are rewarded most for asking difficult questions, independent of their partner's success or failure at responding (*Reward Difficult* game). Students in G_4 are rewarded most for asking easy questions that their partner cannot answer correctly, and are rewarded slightly less for asking difficult questions that their partner can correctly answer (anti-collusive *Teacher's Dilemma* game)[5].

In order to compare observed player strategies, each student was characterized by the relative difficulty (among the seven options) of the majority of the challenges that they selected during their second game[6]. If the majority of words selected were among

[3] The contextual sentences were drawn from an assortment of children's literature in the public domain. Initially, sentences were read aloud and recorded, but in an attempt to rapidly expand the game's problem domain, we began generating recordings using text-to-speech software.

[4] We initially used the *Scrabble*-score of a word as an approximation of difficulty.

[5] The skew in values is meant to prevent player collusion, which is theoretically possible within G_2.

[6] The first game was ignored in order to provide an opportunity to become familiarized with the game.

T_1	E	H
$+$	10	0
$-$	10	0

T_2	E	H
$+$	0	10
$-$	10	0

T_3	E	H
$+$	0	10
$-$	0	10

T_4	E	H
$+$	0	10
$-$	20	10

Figure 3. The Teacher-Matrix used in game-play had different parameter values for each of the four groups in the motivation experiment. The values for *v, j, r,* and *c* (from Figure 1) for the groups are listed here.

	Asks Hard	*Asks Medium*	*Asks Easy*	*Asks Mixed*	*Game Description*
G_1	25%	10%	45%	20%	Reward Easy
G_2	33%	29%	0%	38%	TD
G_3	70%	9%	7%	14%	Reward Difficult
G_4	46%	27%	0%	27%	Anti-collusive TD

Figure 4. Percentages of players within each group that behaved consistent with strategies at top. Each group plays using the correspondingly-numbered TD Teacher-Matrix from Figure 3.

the most difficult two options, the player's strategy was characterized as *Asks Hard*, if the majority were among the middle three options then the player's strategy was *Asks Medium*, and if the majority were among the least difficult two options then the player's strategy was *Asks Easy*. Players without any such majority were characterized as *Asks Mixed*. Figure 4 shows the resulting distributions of observed strategies, by group.

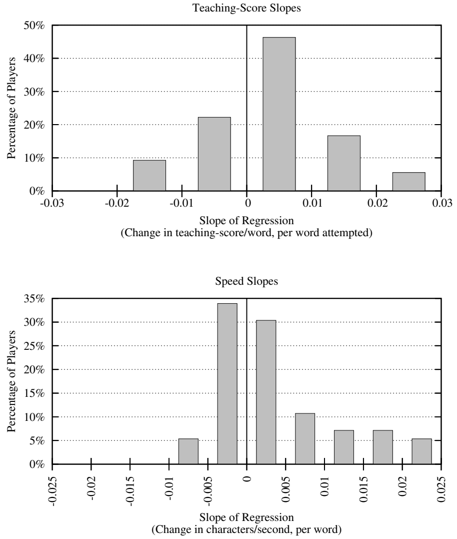
While the resulting variations were less pronounced than expected, they were noticeable. Those playing the *Reward Easy* game chose *Asks Easy* strategies more often than any other group and, similarly, those playing the *Reward Difficult* game chose *Asks Hard* strategies more often than any other group. Those playing the *Teacher's Dilemma* game chose *Asks Mixed* strategies more often than any other groups, which reflects our expected two-pronged strategy. Players in the anti-collusive *Teacher's Dilemma* game slightly less frequently chose *Asks Mixed* strategies, as would be expected from the one-sided bias of their matrix.

After reaching these results with the Spellbee prototype, we selected the G_4 game as the basis for the production version of Spellbee. The remainder of the paper assumes the use of this matrix. While players could theoretically collude to subvert this particular game variation, no such attempt has ever been made by any partner-pairs[7].

5. Observation: On Learning

Identifying and quantifying learning in a system of this sort is inherently difficult. What follows is an admittedly crude attempt to characterize changes in player behavior over time. We examine change with respect to *accuracy, difficulty, speed,* and *teaching-value*, and characterize it based upon the slope of a linear regression of a player's corresponding data, as a crude measure of direction and rate of change. If players are improving, we would expect such slopes to primarily be positive.

[7] Collusion would take the form of both players always selecting the easiest word available and then always responding to challenges incorrectly. In the past year, no player pair has done this for an entire game, or even for a majority of rounds of a game.



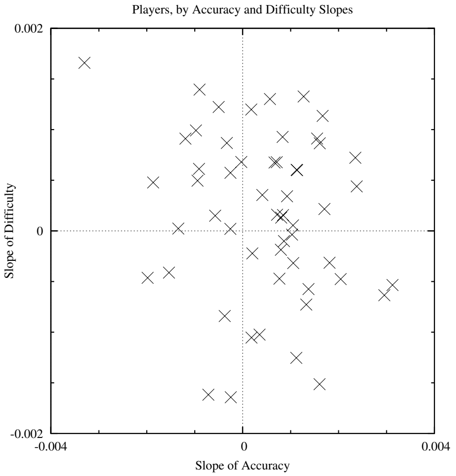


Figure 5. Graphs show players distributed according to the slope of linear regressions of data from the first 20 completed games of play. Anything to the right of the zero line indicates a positive trend.

Figure 6. Players are plotted according to their difficulty slopes and accuracy slopes. Note that very few players occur in the lower-left quadrant.

For this set of experiments, we consider a refined subset of the data collected by the online Spellbee.org system[8]. Of these, we focus only on the first 20 games of players who have completed 20 or more games. Fifty-five players met all of these conditions. Given each player's sequence of 140 rounds of participation (20 games of 7 questions each), we calculate four data points at each round. In Figure 5, *speed* is measured in terms of average number of characters typed per second, and *teaching-score* is the Teacher-Points accrued in that round. In Figure 6, *difficulty* is determined by the New Iowa score for the player's grade-level, and *accuracy* is recorded as a binary correctness value.

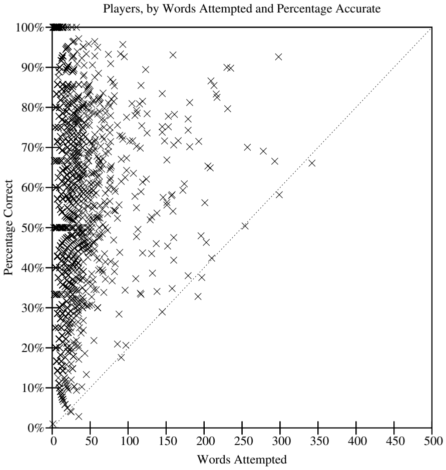
In Figure 6, we graph each player in two dimensions: according to their difficulty and accuracy slopes. This allows us to differentiate among players in the four quadrants (players who do increasingly well or increasingly poor on problems of increasing or decreasing difficulty.) The graphs in Figures 5 and 6 indicate modest changes, but we wish to reiterate that the length of time used for these studies was relatively short. As more students participate for longer periods of time, our analysis can grow accordingly.

6. Discussion

One salient characteristic of open web-based educational systems like Spellbee is that participation is generally voluntary[9]. The non-trivially affects the dynamics of the system, in that the peer-tutoring network is only effective when it is able to retain student

[8] We consider only data recorded during a one-year period (February 1, 2004 through February 1, 2005), only considering players in grades 3-7 (inclusive), and only considering *completed* games (seven rounds finished.)

[9] The exception would be students in classrooms in which the teacher chose to have their class participate.



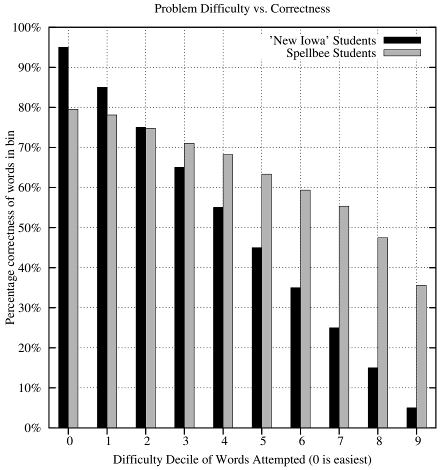


Figure 7. Players are plotted according to the number of words that they attempted while actively using the Spellbee system and the percentage of those words correctly spelled. A dotted line approximates the observed threshold at which students seemed to lose interest or motivation to continue participating.

Figure 8. Words attempted by Spellbee players are classified by difficulty deciles according to the *New Iowa* scale. We then compare the percentage of these words spelled correctly by the students participating in Spellbee as compared to the students participating in the New Iowa study.

interest and participation over time. We seek to maintain this interest purely through the increasingly individualized and engaging educational interactions, rather than through extraneous means[10]. When we began exploring the return-rate data over the past year, we found that the rate of success that a student has at the game (used as an indicator of engagement) provides information about their likelihood of returning. In Figure 7, poorly engaged players (with extremely low rates of spelling accuracy) seem to have a consistent threshold for the maximum amount of repeat participation.

The spelling accuracy data that we have collected with Spellbee can yield the same type of statistics as provided by Greene's *New Iowa Spelling Scale* study [6]. In Figure 8, we compare expected student spelling-accuracy results according to the Iowa metric to the observed results from Spellbee participants. This suggests that we could theoretically stop using the Iowa data in our word-difficulty metric, and replace it with the empirical data that Spellbee have collected to date. While we have not yet taken this step, it suggests an interesting opportunity: when working with a domain for which no readily-available measure of difficulty exists, a rough heuristic can be used initially to bootstrap, and can later be replaced with a metric that uses the empirical data collected thus far.

While we have been leveraging the flexibility and openness of an internet-based system, we continue to encourage and support organized classroom participation. We recently found that one elementary school system in Michigan has over 900 students using Spellbee in school, and we hope to engage in more controlled studies with such groups in the future. This large-scale school-based participation seems particularly notable in

[10] Two frequently-requested additions to the Spellbee system are chat-functionality and a one-player version of the game. We have not implemented any extra-game communication channels due for reasons of child-safety, and we have avoided adding software players to the system in an effort to focus solely on the interpersonal nature of the peer-tutoring network.

light of work by Fishman, et al. suggesting that the adoption of research innovations by schools is often hindered by issues of system-paradigm scalability [5]. The active participation of this school district suggests that reciprocal tutoring networks like Spellbee may be as appropriate as an in-school activity as it has been as an extra-curricular activity.

The motivational layer that we have added to the reciprocal tutoring protocol enables a community of learners to learn to provide each other with the same sorts of appropriate challenges as a teacher may. As participants become more experienced at targeting the challenges that they provide, the tutoring system as a whole has improved as a learning environment. While this adaptive behavior is merely enabled and motivated by our system, this may be sufficient. Leveraging our human-in-the-loop design, we are able to envision tutoring systems that can be easily repurposed from one content domain to another.

7. Acknowledgements

The authors would like to thank members of the DEMO Lab for many useful discussions, in particular Shivakumar Viswanathan for feedback on earlier drafts of this paper. This work was made possible by funding from NSF REC #0113317, the Hewlett Foundation, and the Spencer Foundation.

References

[1] R.S. Baker, A.T. Corbett, and K.R. Koedinger. Detecting student misuse of intelligent tutoring systems. In *Proceedings of the 7th International Conference on Intelligent Tutoring Systems*, pages 43–76, 2004.

[2] Tak-Wai Chan and Chih-Yueh Chou. Exploring the design of computer supports for reciprocal tutoring. *International Journal of Artificial Intelligence in Education*, 8:1–29, 1997.

[3] Li-Jie Chang, Jie-Chi Yang, Tak-Wai Chan, and Fu-Yun Yu. Development and evaluation of multiple competitive activities in a synchronous quiz game system. *Innovations in Education and Teaching International*, 40(1):16–26, 2003.

[4] Mathew Davies and Elizabeth Sklar. Modeling human learning as a cooperative multi agent interaction. In *AAMAS Workshop on Humans and Multi-Agent Systems, at the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2003.

[5] Barry Fishman, Ronald W. Marx, Phyllis Blumenfeld, Joseph Krajcik, and Elliot Soloway. Creating a framework for research on systemic technology innovations. *The Journal of the Learning Sciences*, 13(1):43–76, 2004.

[6] Harry A. Greene. *New Iowa Spelling Scale*. University of Iowa, Iowa City, 1954.

[7] V.S. Kumar, G.I. McCalla, and J.E. Greer. Helping the peer helper. In *Proceedings of the International Conference on AI in Education*, pages 325–332, 1999.

[8] Rikke Magnussen and Morten Misfeldt. Player transformation of educational multiplayer games. In Jonas Heide Smith and Miguel Sicart, editors, *Proceedings of the Other Players Conference*, Copenhagen, Denmark, 2004. IT University of Copenhagen.

[9] Jordan B. Pollack and Alan D. Blair. Co-evolution in the successful learning of backgammon strategy. *Machine Learning*, 32(3):225–240, 1998.

[10] Elizabeth Sklar, Mathew Davies, and Min San Tan Co. Simed: Simulating education as a multi agent system. In N. Jennings, C. Sierra, L. Sonenberg, and M. Tambe, editors, *Third International Conference of Autonomous Agents and Multi-Agent Systems*, 2004.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Do Performance Goals Lead Students to Game the System?

Ryan Shaun BAKER, Ido ROLL, Albert T. CORBETT, Kenneth R. KOEDINGER

Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213 USA

rsbaker, idoroll, corbett, koedinger @cmu.edu

Abstract. Students approach the learning opportunity offered by intelligent tutoring systems with a variety of goals and attitudes. These goals and attitudes can substantially affect students' behavior within the tutor, and how much the student learns. One behavior that has been found to be associated with poorer learning is gaming the system, where a student attempts to complete problems and advance through an educational task by systematically taking advantage of properties and regularities in the system used to complete that task. It has been hypothesized that students game the system because of performance goals. In this paper, however, we find that the frequency of gaming the system does not correlate to a known measure of performance goals; instead, gaming is correlated to disliking computers and the tutor. Performance goals, by contrast, are shown to be associated with working slowly and avoiding errors, and are found to not be correlated to differences in learning outcomes.

1. Introduction

Understanding the student has always been a focus of intelligent tutoring research, but in recent years, there has been a distinct shift in what we are trying to understand about students. In the early years of the field, student modeling focused mostly on issues of knowledge and cognition: modeling what a student knew about the tutor's subject matter, how students acquired and constructed knowledge, and how incorrect knowledge could be modeled and responded to. This research focus led to intelligent tutoring systems that can effectively assess and adapt to students' knowledge about the educational domain, improving learning outcomes [10,17].

In recent years, there has been increasing evidence that students' behavior as they use intelligent tutoring systems is driven by a number of factors other than just their domain knowledge. There is increasing evidence that students with different motivations, beliefs, or goals use tutoring systems and other types of learning environments differently [3,7,9,11]. Furthermore, behaviors that appear to stem from factors other than student knowledge, such as abusing tutor help and feedback [1,6,8] or repeating problems over and over [19], can result in substantially poorer learning outcomes.

While these sorts of findings inform the design of more educationally effective tutors, they are by themselves incomplete. Knowing that a student possesses or fails to possess specific motivations, attitudes, or goals does not immediately tell us whether that student is in need of learning support. Similarly, observing a student using a tutor in a fashion associated with poorer learning does not tell us why that student is choosing to use the tutor in that fashion. If we observe that a specific behavior is associated with poorer learning, we can simply re-design the tutor to eliminate the behavior (cf. [8]), but if the behavior is symptomatic of a broader motivational problem, such a solution may mask the problem rather than eliminate it.

Hence, in order to design systems that can respond to student goals, attitudes, and behaviors in a fashion that positively impacts learning, it is valuable to research all of these factors together. That way, we can learn what motivations, goals, and beliefs lead students to engage in behaviors that negatively impact learning.

2. Gaming the System

In this paper, we apply this combined research approach to the question of why students choose to game the system, a strategy found to be correlated to poorer learning [6]. Gaming the system is behavior aimed at completing problems and advancing through an educational task by systematically taking advantage of properties and regularities in the system used to complete that task, rather than by thinking through the material. In [6], students were observed engaging in two types of gaming the system: systematic trial-and-error, and help abuse, where a student quickly and repeatedly asks for help until the tutor gives the correct answer, often before attempting to solve the problem on his or her own (cf. [1,23]). Within that study, gaming was strongly negatively correlated with learning; students who frequently gamed learned 38% less than students who never gamed, controlling for pre-test score. By contrast, off-task behaviors such as talking to neighbors (about subjects other than the tutor or educational domain) or surfing the web were not negatively correlated with learning. This finding was refined in later analysis, where machine learning determined that gaming students split into two behaviorally distinguishable groups, one which gamed but still learned, and another which gamed and failed to learn [4]. These two groups appeared identical to human observers, but were distinguishable to the machine learning algorithm.

Students who have performance goals, focusing on performing well rather than learning [14], have been found to engage in behaviors that appear similar to gaming, such as seeking answers before trying to solve a problem on their own [2]. For this reason, both our research group [6] and other researchers [18] have hypothesized that students game because of performance goals. A second hypothesis is that students might game out of anxiety, gaming out of the belief that they cannot succeed otherwise [6, cf. 12]. The anxiety hypothesis was supported by evidence that students who game in the harmful fashion tend to game on the hardest steps of the problem [4]. It is also worth noting that having performance goals has been found to lead to anxiety and withdrawal of effort [14] – therefore these two hypotheses may not be inconsistent.

In the remainder of this paper, we will present a study designed to investigate which student goals, beliefs and motivations are associated with gaming the system, with the goal of understanding which of these two hypotheses better explains why students game – or if students game for another reason entirely.

3. Study Methods

We studied student goals, attitudes, behavior, and learning within 6 classes at 2 schools within the Pittsburgh suburbs. All students were participating in a year-long cognitive tutor curriculum for middle school mathematics. Student ages ranged from approximately 12 to 14. 102 students completed all stages of the study; 23 other students were removed from analysis due to missing one or more parts of the study.

We studied these students during the course of a short (2 class period) cognitive tutor lesson on scatterplot generation and interpretation [5]. Within this study, we combined the following sources of data: a questionnaire on student motivations and beliefs, logs of each student's actions within the tutor (analyzed both in raw form, and through a gaming detector (cf. [4]), and pre-test/post-test data. Classroom observations were also obtained in order to improve the gaming detector's accuracy.

The questionnaire consisted of a set of self-report questions given along with the pre-test, in order to assess students' motivations and beliefs. The questionnaire items were drawn from existing motivational inventories or from items used across many prior studies with this age group, and were adapted minimally (for instance, the words "the computer tutor" was regularly substituted for "in class", and questions were changed from first-person to second-person for consistency). All items were pre-tested for comprehensibility with a student from the relevant age group before the study.

The questionnaire included items to assess:

- Whether the student was oriented towards performance or learning (2 items, 4 choices) (e.g. [20])
  "We are considering adding a new feature to the computer tutors, to give you more control over the problems the tutor gives you. If you had your choice, what kind of problems would you like best?
  A) Problems that aren't too hard, so I don't get many wrong.
  B) Problems that are pretty easy, so I'll do well.
  C) Problems that I'm pretty good at, so I can show that I'm smart
  D) Problems that I'll learn a lot from, even if I won't look so smart."
- The student's level of anxiety about using the tutor (2 items, scale 1-6) (e.g. [16])
  "When you start a new problem in the tutor, do you feel afraid that you will do poorly?"
  "When you are working problems in the tutor, do you feel that other students understand the tutor better than you?"
- The student's level of anxiety about using computers (1 item, scale 1-6) (e.g. [16])
  "When you use computers in general, do you feel afraid that you will do something wrong?"
- How much the student liked using the tutor (2 items, scale 1-6) (e.g. [20])
  "How much fun were the math problems in the last computer tutor lesson you used?"
  "How much do you like using the computer tutor to work through math problems?"
- The student's attitude towards computers (1 item, scale 1-6) (e.g. [15])
  "How much do you like using computers, in general?"
- If the student was lying or answering carelessly on the questionnaire. (1 item, 2 choices) (e.g. [21])
  "Is the following statement true about YOU? 'I never worry what other people think of me'. TRUE/FALSE"

Tutor log files were obtained as a source of data on students' actions within the tutor, for a sum total of 30,900 actions across the 106 students. For each action, we distilled 26 features (see [4] for more detail), consisting of:

- Data on how much time the current action (and recent actions) took
- The student's history of errors and help at the current skill and on recent steps
- What type of interface widget was involved in the action
- Whether the action was an error, a bug, correct, or a help request
- The tutor's assessment of the probability that the student knew the skill involved in the action [cf. 10]
- Whether the current action was the first action on the current problem step
- Whether the current problem step involved an "asymptotic" skill that most students knew before starting the tutor, or after the first opportunity to practice it

Using a combination of log files and classroom observations from this study and [6], we trained a gaming detector to assess how frequently a student engaged in harmful gaming and non-harmful gaming [4]. Within the analyses in this paper, we use this gaming detector's assessments as a measure of each student's incidence of harmful and non-harmful gaming rather than direct observations of gaming, for two reasons: First, because our direct observations did not distinguish between harmful gaming and non-harmful gaming whereas the detector could successfully make this distinction – and the two types of gaming may arise from different motivations. Second, because the gaming detector's assessments are more precise than our classroom observations – 2-3 researchers can only obtain a small number of observations of each student's behavior, but the gaming detector can make a prediction about every single student action.

Finally, a pre-test and post-test (the same tests as in [5,6]) were given in order to measure student learning. Two nearly isomorphic problems were used in the tests. Each problem was used as a pre-test for half of the students, and as a post-test for the other half. The tests were scored in terms of how many of the steps of the problem-solving process were correct; in order to get the richest possible assessment of students' knowledge about the material covered in the tutor lesson, the items were designed so that it was often possible to get later steps in the problem correct even after making a mistake.

4. Results

4.1 Gaming The System

Within this study, two types of questionnaire items were found to be significantly correlated to the choice to game: a student's attitude towards computers, and a student's attitude towards the tutor. Students who gamed in the harmful fashion (as assessed by our detector) liked computers significantly less than the other students, F(1,100)=3.94, p=0.05, r = -0.19, and liked the tutor significantly less than the other students, F(1,100)= 4.37, p=0.04, r= -0.20. These two metrics were related to each other: how much a student liked computers was also significantly positively correlated to how much a student liked the tutor, F(1,100)= 11.55, p<0.01, r= 0.32. Gaming in the non-harmful fashion was not correlated to disliking computers, F(1,100) = 1.71, p=0.19, or disliking the tutor, F(1,100)=0.40, p=0.53.

By contrast, our original hypotheses for why students might game did not appear to be upheld by the results of this study. Neither type of gaming was correlated to having performance goals (defined as answering in a performance-oriented fashion on both questionnaire items), F(1,100)=0.78, p=0.38, F(1,100)=0.0,p=0.99. Furthermore, a student's reported level of anxiety about using the tutor was not associated with choosing to game the system, in either fashion, F(1,100) = 0.17, p=0.68, F(1,100) = 1.64, p= 0.20 and a student's reported level of anxiety about using computers was not associated with choosing to game the system, in either fashion, F(1,100)=0.04, p=0.84, F(1,100) = 0.58, p=0.45.

Table 1. Correlations between gaming the system, the post-test (controlling for pre-test), and items on our motivational/attitudinal questionnaire. Statistically significant relationships (p<0.05) are in italics.

	Performance Goals	Anxiety about Using Computers	Anxiety about Using the Tutor	Lying/ Answering Carelessly	Liking Computers	Liking the Tutor
Gaming the System (Harmful fashion)	0.00	-0.02	-0.04	0.06	***- 0.19***	***- 0.20***
Post-Test	0.15	-0.02	0.04	0.03	***-0.32***	0.10

The different types of gaming were associated with learning in a fashion that corresponded to earlier results. Harmful gaming was negatively correlated with post-test score, when controlling for pre-test, F(1,97)=5.61,p=0.02, partial r = -0.33, providing a replication of the finding in [6] that gaming is associated with poorer learning. Additionally, non-harmful gaming did not correlate significantly to post-test score (controlling for pre-test), F(1, 97)= 0.76, p=0.38.

Since harmful gaming is correlated to poorer learning, and harmful gaming is correlated to disliking computers, it is not surprising that a student's attitude towards computers was significantly negatively correlated to their post-test score, F(1,97)=11.51, p<0.01, partial r = - 0.32, controlling for pre-test. To put the size of this effect in context, students who reported disliking computers (i.e. responding 1-2 on the survey item) or being neutral to computers (i.e. responding 3-4) had an average pre-post gain of 18%, whereas students who reported liking computers (i.e. responding 5-6) had an average pre-post gain of 33%. However, the link between computer attitudes and the student's post-test remained significant when harmful gaming (along with pre-test) is partialed out, F(1,96)= 8.48, p<0.01, and the link between harmful gaming and post-test remained significant when computer attitudes (along with pre-test) are partialed out, F(1,96)=3.54, p=0.06. This indicates that, although computer attitudes and gaming are linked, and both are connected to learning, the two have effects independent of each other. By contrast, a student's attitude towards the tutor was not significantly correlated to his/her post-test score, F(1,97) = 0.99, p=0.32, controlling for pre-test.

At this point, our original hypothesis (that gaming stems from performance goals) appears to be disconfirmed. On the other hand, we now know that students who game dislike computers and the tutor – but this raises new questions. Why do students who dislike computers and the tutor game? What aspects of disliking computers and the tutor are associated with gaming?

One possibility is that a student who has a negative attitude towards computers and the tutor may believe that a computer cannot really give educationally helpful hints and feedback – and thus, when the student encounters material she does not understand, she may view gaming as the only option. Alternatively, a student may believe that the computer doesn't care how much he learns, and decide that if the computer doesn't care, he doesn't either. A third possibility is that a student may game as a means of refusing to work with a computer she dislikes, without attracting the teacher's attention. All three of these possibilities are consistent with the results of this study; therefore, fully understanding the link between disliking computers and the tutor and the choice to game the system will require further investigation, probing in depth gaming students' attitudes and beliefs about computers (cf. [15]) and tutors.

4.2 Performance Goals

Entering this study, a primary hypothesis was that performance goals would be associated with a student's choice to game the system. However, as discussed in the previous section, this hypothesis was not upheld: we did not find a connection between whether a student had performance goals and whether that student gamed the system. Instead, performance goals appeared to be connected to a different pattern of behavior: working slowly, and making few errors.

Students with performance goals (defined as answering in a performance goal-oriented fashion on both questionnaire items) answered on tutor problem steps more slowly than the other students, F(1,29276)=39.75, p<0.001, controlling for the student's pre-test score and the student's knowledge of the current tutor step[1]. Overall, the median response time of students with performance goals was around half a second slower than that of the other students (4.4s .vs. 4.9s). Students with performance goals also made fewer errors per problem step than other students, F(1,15854)= 3.51, p=0.06, controlling for the student's pre-test score. Despite having a different pattern of behavior, students with performance goals completed the same number of problem-steps as other students, because slower actions were offset by making fewer errors, t(100)=0.17, p=0.86 (an average of 159 steps were completed by students with performance goals, compared to 155 steps for other students). Similarly, students with performance goals did not perform significantly better or worse on the post-test (controlling for pre-test) than other students, F(1,97)=2.13, p=0.15.

One possible explanation for why students with performance goals worked slowly and avoided errors rather than gaming is that these students may have focused on performance at a different grain-size than we had expected. We had hypothesized that students with performance goals would more specifically have the goal of performing well over the course of days and weeks, by completing more problems than other students – a goal documented in past ethnographic research within cognitive tutor classes [22]. We hypothesized that, in order to realize that goal, students would game the system. However, a student with another type of performance goal might focus on maintaining positive performance minute-by-minute. Such a student would set a goal of continually succeeding at the tutor, avoiding errors and attempting to keep their skill bars continually rising. These students could be expected to respond more slowly than other students, in order to avoid making errors – which is the pattern of behavior we observed.

An alternate account for why students with performance goals may work slowly and avoid errors comes from Elliot and Harackiewicz's 3-goal model of goal-orientation [13], which competes with the 2-goal model that our questionnaire items were drawn from [12]. In both models, students may have learning goals, but where the 2-goal model postulates a single type of performance goal, the 3-goal model states that students with performance goals may have either performance-approach goals (attempting to perform well) or performance-avoidance goals (attempting to avoid performing poorly). The 3-goal model might suggest that the students we identified as having performance goals actually had performance-avoidance goals, and that this was why these students tried to avoid making errors. That explanation would leave as an open question what sort of behavior students with performance-approach goals engaged in. However, in the 3-goal model, students with performance-avoidance goals are also predicted to have anxiety about the learning situation, and there was not a significant correlation between performance goals and tutor anxiety within our data, F(1,100) = 1.52, p=0.22 – suggesting that this questionnaire item was not solely capturing students with performance-avoidance goals.

On the whole, within our study, students with performance goals used the tutor differently than other students, but by working slowly and avoiding errors rather than by gaming the system. It is not yet entirely clear why students with performance goals chose to use the tutor in this fashion – one possible explanation is that these students focused on performance at a different grain-size than expected. In general, it appears that performance goals are not harming student learning, since students with performance goals learned the same amount as the other students. Therefore, recognizing differences in student goals and trying to facilitate a student in his/her goal preferences (cf. [18]) may lead to better educational results than attempting to make all students adopt learning goals.

[1] It is necessary to control for the student's knowledge of the current step for this analysis, since students who make more errors would be expected to have more actions on skills they know poorly – and actions on skills known poorly might be faster or slower in general than well-known skills.

5. Conclusions

The relationships between a student's motivations and attitudes, their actions within a tutoring system, and the learning outcome can be surprising. In this study, we determined that gaming the system, a behavior associated with poor learning, appears to not be associated with having performance goals or anxiety, contrary to earlier predictions. Instead, gaming the system was linked to disliking computers and the tutor. However, we do not yet know how disliking computers and the tutor leads students to game the system; there are several possible explanations for this relationship, from students not believing that the tutor's help and feedback could be educationally helpful, to students using gaming as a means of refusing to work with a computer they dislike. In order to design systems which can respond appropriately when a student games the system, it will be important to develop a richer understanding of the connection between the choice to game, and students' attitudes and beliefs about computers and tutoring systems.

Students with performance goals did not game the system. Instead, these students worked slowly within the tutor and made fewer errors per step than other students. One potential explanation is that students with performance goals focused on performing well at a step-by-step level, rather than attempting to perform well on a longer time-scale through completing more problems than other students. Another possibility is that the students with performance goals in our study more specifically had the desire to avoid performing poorly (cf. [13]), but this explanation is inconsistent with the lack of significant correlation between performance goals and anxiety.

One other question for future work is how well the findings presented here will generalize to other educational contexts. In this paper, we studied the links between motivations/attitudes, behavior within the tutor, and learning within the context of 12-14 year old students, who use cognitive tutors as part of a full-year curriculum, in public school classrooms in the suburban northeastern United States. It is quite possible that the relationships between students' motivations/attitudes, behavior within the tutor, and learning will differ across settings and populations.

Nonetheless, the results of this study demonstrate the value of combining data about how individual students use tutors with motivational, attitudinal, and learning data. In order to design tutors that can adapt to students in a fashion that improves learning, we need to know what behaviors are associated with poorer learning, and why students engage in these behaviors. The answers to these questions can be non-intuitive: before [6], we did not expect gaming the system to be the behavior most strongly connected with poor learning; before this study, we did not expect computer and tutor attitudes to be the best predictors of gaming. However, with this information in hand, we can now focus our efforts towards designing remediations for gaming (as opposed to other behaviors), and do so in a fashion that takes into account what we know about why students choose to game (as opposed to simply trying to prevent gaming, or using an incorrect hypothesis for why students game) – improving our chances of designing intelligent tutors that can guide all students to positive educational outcomes.

Acknowledgements

We would like to thank Angela Wagner, Jay Raspat, Megan Naim, Sue Cameron, Russ Hall, Dina Crimone and Frances Battaglia for assistance in conducting the studies at the schools, and Vincent Aleven, Joseph Beck, Cecily Heiner, Steve Ritter, and Carolyn Rosé for helpful discussions and suggestions. This work was funded by an NDSEG (National Defense Science and Engineering Graduate) Fellowship, and by NSF grant REC-043779 to

"IERI: Learning-Oriented Dialogs in Cognitive Tutors: Toward a Scalable Solution to Performance Orientation".

References

[1] Aleven, V., Koedinger, K.R. Investigations into Help Seeking and Learning with a Cognitive Tutor. In R. Luckin (Ed.), Papers of the AIED-2001 Workshop on Help Provision and Help Seeking in Interactive Learning Environments (2001) 47-58

[2] Arbreton, A. (1998) Student Goal Orientation and Help-Seeking Strategy Use, In S.A. Karabenick (Ed.), *Strategic Help Seeking: Implications for Learning and Teaching.* Mahwah, NJ: Lawrence Erlbaum Associates, 95-116.

[3] Arroyo, I., Murrary, T., Woolf, B.P. (2004) Inferring Unobservable Learning Variables From Students' Help Seeking Behavior. *Proceedings of the Workshop on Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes, at the 7th International Conference on Intelligent Tutoring Systems*, 29-38

[4] Baker, R.S., Corbett, A.T., and Koedinger, K.R. (2004a) Detecting Student Misuse of Intelligent Tutoring Systems. *Proceedings of the 7th International Conference on Intelligent Tutoring Systems,* 531-540.

[5] Baker, R.S., Corbett, A.T., and Koedinger, K.R. (2004b) Learning to Distinguish Between Representations of Data: a Cognitive Tutor That Uses Contrasting Cases. *Proceedings of the International Conference of the Learning Sciences,* 58-65.

[6] Baker, R.S., Corbett, A.T., Koedinger, K.R., and Wagner, A.Z. (2004c) Off-Task Behavior in the Cognitive Tutor Classroom: When Students "Game the System". Proceedings of ACM CHI 2004: Computer-Human Interaction, 383-390.

[7] Bartholomé, T., Stahl, E., & Bromme, R. (2004). Help-seeking in interactive learning environments: Effectiveness of help and learner-related factors in a dyadic setting. *Proceedings of the International Conference of the Learning Sciences: Embracing diversity in the learning sciences*, 81-88.

[8] Beck, J. (2004) Using Response Times to Model Student Disengagement. *Proceedings of the ITS2004 Workshop on Social and Emotional Intelligence in Learning Environments*, 13-20.

[9] Conati, C., Maclaren, H. (2004) Evaluating a Probabalistic Model of Student Affect. *Proceedings of the 7th International Conference on Intelligent Tutoring Systems,* 55-64.

[10] Corbett, A.T., Anderson, J.R. (1995) Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *User Modeling and User-Adapted Interaction,* 4, 253-278.

[11] deVicente, A., Pain, H. (2002) Informing the Detection of the Students' Motivational State: An Empirical Study. *Proceedings of the Sixth International Conference of Intelligent Tutoring Systems,* 933-943.

[12] Dweck, C.S. (1975) The Role of Expectations and Attributions in the Alleviation of Learned Helplessness. *Journal of Personality and Social Psychology,* 54 (1), 674-685.

[13] Elliot, A.J., Harackiewicz, J.M. (1996) Approach and Avoidance Achievement Goals and Intrinsic Motivation: A Mediational Analysis. *Journal of Personality and Social Psychology,* 70 (3), 461-475.

[14] Elliot, E.S., Dweck, C.S. (1988) Goals: An Approach to Motivation and Achievement. *Journal of Personality and Social Psychology,* 31 (4), 674-685.

[15] Frantom, C.G., Green, K.E., Hoffman, E.R. (2002) Measure Development: The Children's Attitudes Toward Technology Scale (CATS). *Journal of Educational Computing Research,* 26 (3), 249-263.

[16] Harnisch, D.L., Hill, K.T., Fyans, L.J. (1980) Development of a Shorter, More Reliable, and More Valid Measure of Test Motivation. Paper presented at the 1980 annual meeting of the National Council on Measurement in Education. ERIC Document # ED193273.

[17] Martin, J., vanLehn, K. Student Assessment Using Bayesian Nets. *International Journal of Human-Computer Studies,* 42 (1995), 575-591.

[18] Martínez Mirón, E.A., du Boulay, B., Luckin, R. (2004) Goal Achievement Orientation in the Design of an ILE. *Proceedings of the ITS2004 Workshop on Social and Emotional Intelligence in Learning Environments*, 72-78.

[19] Mostow, J., Aist, G., Beck, J., Chalasani, R., Cuneo, A., Jia, P., Kadaru, K. A La Recherche du Temps Perdu, or As Time Goes By: Where Does the Time Go in a Reading Tutor that Listens? Sixth International Conference on Intelligent Tutoring Systems (2002) 320-329

[20] Mueller, C.M., Dweck, C.S. (1998) Praise for Intelligence Can Undermine Children's Motivation and Performance. *Journal of Personality and Social Psychology,* 75 (1), 33-52.

[21] Sarason, S.B. (1978) *Anxiety in Elementary School Children: A Report of Research.* Westport, CT: Greenwood Press.

[22] Schofield, J.W. (1995) *Computers and Classroom Culture.* Cambridge, UK: Cambridge University Press.

[23] Wood, H., Wood, D. (1999) Help Seeking, Learning, and Contingent Tutoring. *Computers and Education,* 33, 153-159.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Pedagogical agents as social models for engineering: The influence of agent appearance on female choice

Amy L. BAYLOR
Director, Center for Research of Innovative Technologies for Learning (RITL)
http://ritl.fsu.edu
Florida State University
baylor@coe.fsu.edu

E. Ashby PLANT
Department of Psychology
Florida State University
plant@psy.fsu.edu

Abstract. The current work examined the influence of pedagogical agents as social models to increase females' interest in engineering. Seventy-nine female undergraduate students rated pedagogical agents on a series of factors (e.g., most like themselves, most like an engineer, and most prefer to learn from). The agents were identical with the exception of differing by appearance/image in four aspects (age, gender, attractiveness, "coolness"). After selecting the agent from which they most preferred to learn, participants interacted with it for approximately 15 minutes and received a persuasive message about engineering. Results indicated that the women were more likely to choose a female, attractive, young, and cool agent as most like themselves and the one they most wanted to be like. However, they tended to select male, older, uncool agents as the most like engineers and tended to choose to learn about engineering from agents that were male and attractive, but uncool. Interacting with an agent had a positive impact on math-related beliefs. Specifically, the women reported more positive math and science related beliefs compared to their attitudes at the beginning of the semester and compared to a group of women who did not interact with an agent. Further, among the women who viewed an agent, the older version of the agent had a stronger positive influence on their math-related beliefs than the younger agent.

Introduction

Many females possess negative and unconstructive beliefs regarding engineering, both as an occupation in general and as a possible career. These misperceptions are instilled by a social fabric that pervades our society, represented not only within our educational systems but also in homes, within families, and in popular culture [1]. This perceptual framework generally stereotypes engineering and scientific fields as physically challenging, unfeminine, and aggressive [2] as well as object-oriented [3, 4]. As such, these beliefs have implications for how women perceive themselves and their competencies within the engineering and scientific realms.

As early as elementary age, females underestimate their math ability, even though their actual performance may be equivalent to that of same-aged boys [5, 6]. In addition, young females believe that math and engineering aptitudes are fixed abilities, attributing success or failure to extrinsic instead of intrinsic factors [7]. The extent of such gender-differentiating attitudes helps to explain the lower probability of women's completing an engineering or science related program and subsequently choosing other fields where interpersonal and organizational-related aspects have greater emphasis [8].

In order to change women's negative attitudes regarding engineering and science-related fields, it may be possible to use pedagogical agents as mechanism for persuasion. Extensive research [9] has demonstrated that people tend to apply human social rules to computer technologies. Females are particularly influenced by the communication and relational aspect of pedagogical agents and are more influenced by them than males [e.g.,10]. While empirical evidence has shown that pedagogical agents are effective to influence the transfer of learning [11, 12], metacognition [13] and motivation [14-17], there is limited evidence of their effectiveness as social models to influence attitudinal beliefs.

The purpose of this study is to investigate whether pedagogical agents can be used as social models [18] to influence college-age women's attitudes and beliefs about engineering. Of particular interest is the impact of the agents' appearance (or image) for its effectiveness. Research in social psychology would suggest that several appearance features are critical in determining how persuasive a social model would be in influencing young women's engineering-beliefs: age, gender, attractiveness, and coolness [18-20]. In general, people are more persuaded by models that are similar to them or similar to how they would like to be [21-24]. Therefore, agents who are young, female, attractive, and "cool" may be more influential in influencing young women's attitudes. However, people are also more persuaded by those they perceive as experts. Thus, agents who are older and seem to be like the typical or stereotypical engineer (i.e., male and uncool) may be particularly influential.

1. Research Questions

In this study, we address the following research questions:

- Which appearance-related agent features (agent age, gender, attractiveness, "coolness") do females choose in response to questions regarding respect, identification, wanting to be like, engineering-likeness, and serving as an instructor?
- What is the impact of the agent that participants choose to learn from on their mathematics beliefs?

2. Method

2.1 Participants

The sample consisted of 79 female students enrolled in eleven sections of an "Introduction to Educational Technology" course at a Southeastern public university. Participation in this study was a required activity for class participants, and they received course credit for participating. The mean age of the sample was 19.34 (*SD* = 1.41) years old. Eighty percent of the participants were White, 9% were Black/African American, 9% were Hispanic/Latino, 1% were Asian/Asian American, and 1% were Caribbean.

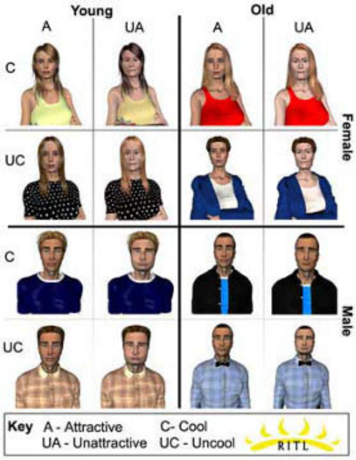
2.2 Materials

Pre-survey. The pre-survey assessed dependent variables in the areas of science/math: identity, utility, interest (as a major and as a job), current and future efficacy, engagement, and future interest. In addition, it included a scale assessing the participants' general self-esteem.

Post-survey. The post survey included all items from the pre-survey in addition to items regarding agent perceptions (e.g., competent, believable, helpful).

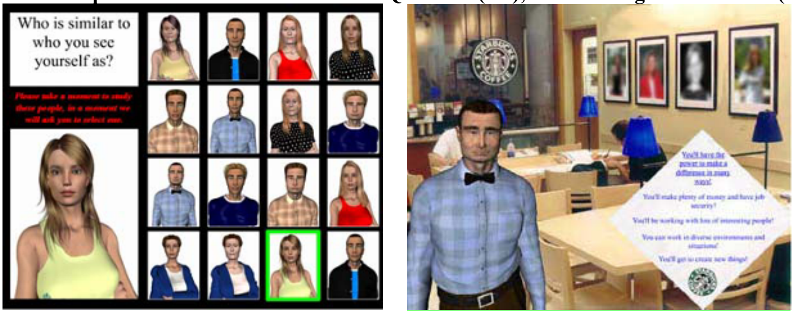
Agents. The agents (see Figure 1) were designed and previously validated to represent 4 different factors (gender: male, female; age: older (~45 years), younger (~25 years); attractiveness: attractive, unattractive; and "coolness:" cool, uncool). Attractiveness was operationalized to include only the agent's facial features, whereas "coolness" included the agent's type of clothing and hairstyle. For example, both of the young attractive female agents have identical faces, but differ in "coolness" by their dress and hairstyle. The agents were created in Poser3D. One male and one female voice were recorded for all the agents using the same script. The audio files were synchronized with the agents using Mimic2Pro. A single series of gestures was added to the agents to complete the agent animation process. A fully integrated environment was created using Flash MX Professional 2004, which allowed for a web browser presentation.

Figure 1. Validated Agents, differing by Age, Gender, Attractiveness, and "Coolness"



Research environment. In the first phase, the participant answered the following series of questions while being presented with the set of 16 agents (see Figure 2): "Who would you most respect and look up to?" "Who would you most want to be like?" "Who is similar to who you see yourself as now?" "Who most looks like an engineer?" "Who looks least like an engineer?"

Figure 2. Sample Screenshots. Phase 1 - Choice Questions (left); Phase 2 - Agent interaction (right)



Who is similar to who you see yourself as?

Please take a moment to study these people, in a moment we will ask you to select one.

You'll have the power to make a difference in many ways!

You'll make plenty of money and have job security!

You'll be working with lots of interesting people!

You can work in diverse environments and situations!

You'll get to create new things!

The agents were randomly presented in one of four combinations that varied the screen layout of the agents to guard against agent selection based on location on the screen (e.g. participants choosing the middle agent). To encourage the participants to give thought to their answer, the participants could not make their choice before 10 seconds had passed. Participants could roll over each agent headshot to see a larger image of the agent. Participants confirmed their selection before proceeding to the next question. The final question "Who would you like to learn from about engineering?" determined which agent presented the persuasive message about engineering in the second phase.

In the second phase, the chosen agent (set in a coffee shop location) introduced itself and provided an approximately ten-minute narrative about four prominent female engineers, followed by five benefits of engineering careers. This script was validated as effective in Baylor & Plant (2004). Periodically, the participants were asked to click on the screen to continue the presentation. Regardless of the participant selection, the agent had identical message and animation.

2.3 Measures

Each dependent variable (with the exception of self-esteem and agent perceptions) was assessed separately for both math and science. Reliability for all scales as assessed by Cronbach's alpha was >.7.

- Identity: three 5-point Likert scale items
- Utility: four 7-point Likert scale items
- Interest (as a major and as a job): three 7-point Likert scale items
- Efficacy : five 5-point Likert scale items
- Engagement : three 7-point Likert scale items
- Self-Esteem : ten 4-point Likert scale items

2.4 Procedure

The pre-survey was distributed at the beginning of the semester. The survey took approximately fifteen minutes to complete. Near the end of the semester, participants accessed the online module through a web-browser during a regularly-scheduled classroom lab session. Following completion, participants answered the post-survey questions (with an image of the agent as a reminder). The whole session took approximately thirty minutes.

2.5 Data analysis and Design

To determine which agent participants chose, based on the six social model characteristics/questions, four one-sample t-tests were conducted for each of the questions to explore whether the female participants' choices were influenced by the gender, age, coolness and attractiveness of the agents.

Given that the agents that participants chose to learn from were primary male, attractive and uncool, the analysis of agent impact was limited to agent age. The six key outcome measures were organized into four conceptually-related categories: identity/engagement, future interest (job or major), efficacy and utility, and were analysed separately. The impact of agent age on future interest and identity/engagement in mathematics were analyzed through two separate one-factor (age: young, old) MANOVAs. Two separate independent sample t-tests were conducted to assess the impact of chosen agent age (young, old) on math self-efficacy and math utility.

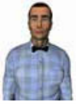
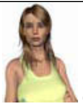
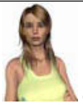
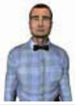
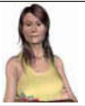
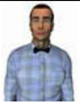
3. Results

Results are organized with respect to each of the two research questions.

3.1 Which appearance-related agent features (agent age, gender, attractiveness, "coolness") do females choose, according to social model characteristic (respect, identification, want to be like, engineering-likeness, and serving as an instructor)?

Four one sample t-tests were performed for each of the 6 questions and results are summarized in Table 1.

Table 1. Summary of Choices by Question and Agent Appearance

	Gender	***Age***	***Attractiveness***	***Coolness***		***Representative Agent (% selected)***
Who would you most respect and look up to?			**Attractive** > Unattractive (67% vs. 33%) **	**Uncool** > Cool (81% vs. 19%) ***		 (16%)
Who would you most want to be like?	**Female** > Male (79% vs. 21%) ***	**Young** > Old (85% vs. 15%) ***	**Attractive** > Unattractive (94% vs. 6%) ***	**Cool** > Uncool (79% vs. 21%) ***		 (72%)
Who is similar to who you see yourself as?	**Female** > Male (81% vs. 19%) ***	**Young** > Old (81% vs.19%) ***	**Attractive** > Unattractive (85% vs. 15%) ***	**Cool** > Uncool (71% vs. 29%) ***		 (53%)
Who most looks like an engineer?	**Male** > female (94% vs. 6%) *	**Old** > Young (63% vs. 37%) ***		**Uncool** > Cool (75% vs. 25%) ***		 (28%)
Who looks least like an engineer?	**Female** > Male (73% vs. 27%) ***	**Young** > Old (69% vs. 31%) **		**Cool** > Uncool (84% vs. 16%) ***		 (24%)
Who would you like to learn from about engineering?	**Male** > female (87% vs. 13%) ***		**Attractive** > Unattractive (69% vs. 31%),**	**Uncool** > Cool (64% vs. 36%) **		 (22%)

**p<.05; **p<.01; ***p<.001*

As shown above in Table 1, female participants tended to: 1) most respect agents that were attractive and uncool; 2) want to be like and 3) identify most with the agent that was female, young, attractive, and cool; 4) find that the older, uncool male agents looked most like an engineer, whereas the young cool females looked the least like engineers; and 5) want to learn from the male agents who were attractive and uncool.

3.2 What is the impact of the agent from which participants choose to learn?

Regardless which agent was chosen to deliver the message (i.e., the agent they selected to "learn from"), following the agent's message, women had significantly more interest in hard sciences as a job (p<.01), more efficacy in math (p<.10), could identify more with the hard sciences (p<.10), more engagement in the hard sciences (p<.05), more future interest in the hard sciences (p<.01), and believed hard sciences was more useful (p<.001) than prior in the semester.

In addition, the responses of the female participants who interacted with an agent were compared to a group of female participants who only completed the post-survey at the end of the semester (N=12). Compared to the group who simply completed the post-survey, the participants who viewed an agent had higher levels of math self-efficacy (*p*<.05), math identity (*p*<.05), math utility (*p*<.01), and future interest in a job in mathmatics (p <.05) at the end of the semester. In addition, they reported a higher general self-esteem (*p*<.001) than the no-agent group.

For the final question ("who would you like to learn from"), participants tended to select male agents that were attractive and uncool, but differing by age (young or old). Consequently, agent impact was limited to comparing the effects of agent influence by age (younger versus older).

The MANOVA for *future interest in math* indicated that there was an overall effect of the age of the agent on the future interest in math, Wilks's Lambda = .917, *F*(2,76)=3.449, *p*<.05. Univariate results revealed a main effect of agent age on future interest in math as a *major*, where those influenced by the older agent reported significantly more future interest in math as a major (*M* = -.663, *SD* = 2.258) compared to participants who had a younger agent (*M* = -1.712, *SD* = 1.677), *F*(1,79)=5.096, MSE = 4.150, p < .05. The effect size estimate is *d* = -.53 indicating a medium effect. Univariate results also revealed a main effect for the agent age on future interest in math as a *job*, where participants who learned from the older agent reported greater future interest in math as a job (*M* = .0435, *SD* = 1.632) compared to participants who had a younger agent (*M* = -.8485, *SD* = 1.253), *F*(1,79)=6.918, MSE = 2.210, *p* = .01. The effect size estimate is *d* = -.61, indicating a medium effect.

The MANOVA for *math identity and engagement* indicated that there was as overall effect of the age of the agent on future interest in math, Wilks's Lambda = .921, *F*(2,76)=3.271, *p*<.05. Univariate results revealed a main effect for the agent age on *math identity*, indicating that participants who learned from an older agent reported a higher level of math identity (*M* = .4783, *SD* = 1.216) than participants who learned from a younger agent (*M* = -.202, *SD* = 1.193), *F*(1,79)=6.106, MSE = 1.456, *p*< .05. The effect size estimate was *d* = -.57, indicating a medium effect. Univariate results also revealed a main effect for the agent age on *math engagement*, indicating that participants who had an older agent reported higher level of math engagement (*M* = .4638, *SD* = 1.856) compared to participants who had a younger agent (*M* = -.5859, *SD* = 1.848), *F*(1,79)=6.167, MSE = 3.433, *p* < .05. The effect size estimate is *d* = -.57, indicating a medium effect.

An independent sample *t*-test revealed that participants who selected an older agent reported higher levels of *math efficacy* compared to participants who had a younger agent. (*M* = .6304 vs. *M* = .1333), *t*(77)=-1.919, *p* = .05. The effect size estimate is *d* = .45, indicating a medium effect. An independent sample *t*-test revealed that participants who selected an older agent reported higher levels of *math utility* compared to participants who had a young agent (*M*= 1.03 vs. *M*=.52), *t* (77) =-1.72, *p*=.05. The effect size estimate is *d* =.40, indicating a medium effect.

These findings indicate that participants who learned from the older agents were more strongly influenced than those who learned from the younger agents. It may be that because the older agents were perceived as more like engineers, as indicated by the participants' ratings at the beginning of the session, they were more effective models. Interestingly, whereas participants were more influenced by the older agents and rated them as more competent than the younger agents, they also rated the younger versions as more believable (p<.1) and helpful (p<.1) than the older ones.

4. Discussion

The findings from the current study indicate that pedagogical agents may be useful tools for modelling positive attitudes toward engineering to young women. In general, the women who interacted with a pedagogical agent developed more positive math and science related beliefs compared to their ratings earlier in the semester as well as compared to a group of young women from the same course who did not interact with an agent. In addition, the present study provided insight into the types of agents that women choose to learn from and the types of agents that were more effective in influencing the women's attitudes regarding math and engineering.

Previous work examining social modelling would indicate that the young women should be more influenced by agents that were similar to them or similar to how they would like to be (e.g., female, attractive, cool). However, persuaders who are perceived as knowledgeable and experts can also be highly influential. As anticipated, when the young women in the current study were asked to select the agents who were most like them and who they most wanted to be like, they tended to pick young, female, attractive, and cool agents. However, they also selected the young, female, cool agents as being least like an engineer. When asked to select who they would most like to learn from about engineering, the women in the current study were far more likely to pick male agents who were uncool but attractive. Interestingly, it was also the male, uncool agents that they tended to rate as most like an engineer. However, their selections for the most typical engineer also tended to be older.

Because so few of the participants selected female agents (only 13%), it was difficult to compare the efficacy of the female compared to male agents. In addition, there was a strong tendency to select attractive, uncool agents from whom to learn. Therefore, it is difficult to pit the efficacy of a similar agent (i.e., young female, attractive, cool) against the efficacy of an agent perceived as an expert on the topic (i.e., stereotypical engineer – male, old, uncool). In order to examine this issue more thoroughly, it will be important in future work to conduct studies where young women are randomly assigned to various agents. However, because the women's choice of agent from whom to learn varied by age, it was possible to explore whether the older or younger agents were more effective. Counter to the idea that similar agents would be more effective, the young women who selected and viewed the *older* compared to younger agents had more future interest in mathematics, greater self-efficacy in mathematics, were more engaged and identified with mathematics, and saw mathematics as having more utility.

Although these findings would seem to suggest that similarity is not as influential as expertise, it is important to note that the agents talked about four prominent female engineers who varied in age. Thus, the impact of hearing the older, therefore, perhaps more typical engineer agent discuss young and old successful female engineers may have constituted a particularly effective persuasive tool.

This study adds to the growing empirical evidence of the importance of interface agent *appearance* [25]. It is important to note that the pedagogical agents in this study were intentionally scripted to control for message, interactivity, animation, and expression. Future research must also consider the additive effects of other important agent persona features (e.g., voice, message, animation), particularly as they serve as front-ends to intelligent tutoring systems that influence attitude and other learning-related outcomes.

5. Acknowledgments

This work was supported by the National Science Foundation, Grant HRD-0429647.

References

[1] C. Muller, "The Under-Representation of Women in Engineering and Related Sciences: Pursuing Two Complimentary Paths to Parity.," presented at National Academies' Government University Industry Research Roundtable Pan-Organizational Summit on the U.S. Science and Engineering Workforce, 2002.

[2] Adams, "Are Traditional Female Stereotypes Extinct at Georgia Tech?" *Focus*, pp. 15, 2001.

[3] G. H. Dunteman, Wisenbaker, J., & Taylor, M.E., "Race and sex differences in college science program participation.," Research Triangle Institute, Research Triangle Park, NC, Report to the National Science Foundation 1978.

[4] R. Lippa, "Gender-related differences and the structure of vocational interests: The importance of the people-things dimension," *Journal of Personality and Social Psychology*, vol. 74, pp. 996-1009, 1998.

[5] J. S. Eccles, "Gender Roles and women's achievement-related decisions," *Psychology of Women Quarterly*, vol. 11, pp. 135-172, 1987.

[6] J. S. Eccles, "Understanding women's educational and occupational choices: Applying the Eccles et al. model of achievement related choices," *Psychology of Women Quarterly*, vol. 18, pp. 585-609, 1994.

[7] G. D. Heyman, Martyna, B. and Bhatia, S., "Gender and Achievement Related Beliefs Among Engineering Students," *Journal of Women and Minorities in Science and Engineering*, vol. 8, pp. 41-52, 2002.

[8] E. Seymour and N. Hewitt, *Talking About Leaving: Why Undergraduates Leave the Sciences*. Boulder, CO: Westview Press, 1997.

[9] B. N. Reeves, C., *The Media Equation*. Stanford, CA: CSLI Publications, 1996.

[10] A. L. Baylor, S. Kim, C. Son, and M. Lee, "The Impact of Pedagogical Agent Emotive Expression and Deictic Gestures on Attitudinal and Procedural Learning Outcomes," presented at AI-ED, Amsterdam, 2005.

[11] R. K. Atkinson, "Optimizing learning from examples using animated pedagogical agents," *Journal of Educational Psychology*, vol. 94, pp. 416-427, 2002.

[12] R. Moreno, Mayer, R.E., Spires, H.A., & Lester, J.C., "The case for social agency in computer-based teaching: do students learn more deeply when they interact with animated pedagogical agents?" *Cognition and Instruction*, vol. 19, pp. 177-213, 2001.

[13] A. L. Baylor, "Expanding preservice teachers' metacognitive awareness of instructional planning through pedagogical agents," *Educational Technology, Research & Development*, vol. 50, pp. 5-22, 2002b.

[14] A. L. Baylor, & Kim, Y., "The Role of Gender and Ethnicity in Pedagogical Agent Perception," presented at the E-Learn World Conference on E-Learning in Corporate, Government, Healthcare & Higher Education, Phoenix, Arizona, 2003a.

[15] A. L. Baylor, & Kim, Y., "Validating Pedagogical Agent Roles: Expert, Motivator, and Mentor," presented at the ED-MEDIA, Honolulu, Hawaii, 2003b.

[16] A. L. Baylor, Shen, E., & Huang, X., "Which Pedagogical Agent do Learners Choose? The Effects of Gender and Ethnicity," presented at the E-Learn World Conference on E-Learning in Corporate, Government, Healthcare, & Higher Education, Phoenix, Arizona, 2003.

[17] Y. Kim, Baylor, A.L., Reed, G., "The Impact of Image and Voice with Pedagogical Agents.," presented at the E-Learn World Conference on E-Learning in Corporate, Government, Healthcare & Higher Education, Phoenix, Arizona, 2003.

[18] A. Bandura, *Self-Efficacy: The Exercise of Control*. New York, New York: W.H. Freeman and Company, 1997.

[19] S. Chaiken, "Communicator physical attractiveness and persuasion," *Journal of Personality and Social Psychology*, vol. 37, pp. 1387-1397, 1979.

[20] B. McIntyer, Paulson, R.M. & Lord, C.G., "Alleviating women's mathematics stereotype threat salience of group achievements," *Journal of Experimental Social Psychology*, vol. 74, pp. 996-1009, 1998.

[21] A. Bandura, *Social Foundations of Thought and Action: A Social Cognitive Theory*. Englewood Cliffs, N.J.: Prentice-Hall, 1986.

[22] T. Mussweiler, "Comparison Processes in Social Judgment: Mechanisms and Consequences," *Psychological Review*, vol. 110, pp. 472-489, 2003.

[23] D. H. Schunk, "Peer Models and Children's Behavioral Change," *Review of Educational Research*, vol. 57, pp. 149-174, 1987.

[24] J. V. Wood, "Theory and Research Concerning Social Comparisons of Personal Attributes," *Psychological Bulletin*, vol. 106, pp. 231-248, 1989.

[25] A. L. Baylor, "The Impact of Pedagogical Agent Image on Affective Outcomes," presented at Intelligent User Interface International Conference, San Diego, CA., 2005.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

The impact of frustration-mitigating messages delivered by an interface agent

Amy L. BAYLOR
Director, Center for Research of Innovative Technologies for Learning (RITL)
http://ritl.fsu.edu
Florida State University
baylor@coe.fsu.edu

Daniel WARREN
Instructional Systems Program RITL – Affective Computing
http://ritl.fsu.edu
Florida State University
rdw4048@fsu.edu

Sanghoon PARK
Instructional Systems Program RITL – Affective Computing
http://ritl.fsu.edu
Florida State University
ssp5177@fsu.edu

E SHEN
Instructional Systems Program RITL – Affective Computing
http://ritl.fsu.edu
Florida State University
ess0086@fsu.edu

Roberto PEREZ
Instructional Systems Program RITL – Affective Computing
http://ritl.fsu.edu
Florida State University
rgp6722@mailer.fsu.edu

Abstract. Mitigating frustration is important within computer-based learning contexts. In this experimental study where participants were purposefully frustrated, the interface agent message (apologetic, empathetic, or silent) was manipulated to investigate its impact on student attitude toward the task, attitude toward the agent, and attribution toward the cause of frustration. Fifty-seven undergraduate students responded to an invitation to participate in a web-based survey and to receive a movie ticket for their effort. An animated interface agent, "Survey Sam," was present as students answered survey items and were confronted with a frustrating obstacle – an error message pop-up window that blocked them from answering the survey items. Survey Sam delivered either an affective message (apologetic or empathetic) or remained silent to the thirty students who actually completed the survey. Results revealed that the presence of an affective message (either apologetic or empathetic) led participants to report significantly greater frustration, suggesting that the affective message reinforced and validated their frustration. However, and more importantly, they attributed the cause of their frustration to the program instead of to themselves (as did the no message group). A comparison of message type (apologetic or empathetic) indicated that participants receiving the empathetic message rated Survey Sam as significantly more believable and sincere. Implications of these findings as a catalyst for further research in the development of frustration-mitigating support for computer-based contexts are discussed.

Introduction

Emotions within learning contexts are not stable. Students may experience many different emotional states during the learning process. According to appraisal theories of emotion, emotions arise from an individual's meaning construction and appraisal of continuous interactions with the world [1, 2]. Especially in learning situations, the process of students' meaning construction and appraisal may acquire different forms depending on the characteristics of the tasks given to those students. Frustration, where an obstacle prevents the satisfaction of a desire [3], is one of the negative emotions students deal with in most learning situations because a learning task usually requires student effort to solve challenging problems. Therefore, reducing the level of frustration becomes a critical issue in a computer-based learning situation [4].

One method for diffusing frustration involves offering an *apology*, especially if the one apologizing is taking responsibility for the obstacle causing the frustration, thus admitting blameworthiness and regret for an undesirable event [5, 6]. A second method to diffuse frustration involves delivering *empathetic concern* for another's emotional experiences, especially if the one expressing concern is not perceived as the cause of the frustration. Empathy is an emotive-cognitive state where the emotional element involves concern with the personal distress of another person and the cognitive element involves understanding the perspective of the other person [7], resulting in a shared, or distributed, emotional experience.

With regard to previous agent implementations, Mori and colleagues evaluated an affective agent that was designed to alleviate frustration during a mathematics quiz game by delivering empathetic "happy for" or "sorry for" responses [8]; however, results were limited by a small sample size. While Johnson and colleagues have found that agent politeness is valuable in a tutoring environment [9], they have not focused on learner frustration. Baylor and colleagues investigated the role of interface agent message (presence/absence of motivation) and affective state (positive versus evasive) on student attitude for mathematically-anxious students [10]. While their results supported the value of cognitively-focused motivational messages [e.g., 11] on student confidence, results were inconclusive regarding the impact of affect as a mediator in the process.

1. Research Questions

This exploratory, experimental study was designed to investigate the impact of interface agent message (apologetic, empathetic, or none) on user frustration, attribution perception, and attitudes. Specifically, we investigated the following research questions:

1. Does the presence of an affective message impact participant attitude toward the task, attitude toward the agent, or attribution toward the cause of frustration?
2. Does the type of affective message (apologetic or empathetic) impact participant attitude toward the task, attitude toward the agent, or attribution toward the cause of frustration?

2. Method

2.1 Participants

Participants included thirty undergraduate students (average age = 19.7 years; 93% female) who had recently completed an introductory course on Educational Technology in a public university in the Southeastern United States. Fifty-five participants began the study, but only thirty actually completed it. Computer self-efficacy assessed as part of the pre-survey revealed no differences in efficacy between those who completed the survey and those who did not, or between treatment groups.

2.2 Research Environment and Interface Agent

The research environment was created to so that participants could complete a personality survey (based on the Big Five Factor theory of personality [e.g.,12] with the presence of "Survey Sam," a 3D animated interface agent. Upon entering the environment, Survey Sam introduced students to the survey, stating: "Hi, my name is Survey Sam. Here's the survey you take to get your movie tickets. Please do your best." While students were completing the survey, Survey Sam was always present and displayed basic animations, including eyeblinking and head-turning, figuratively "watching" participants as they worked through the survey. His presence was maintained throughout the survey to establish his existence as a foundation for the message that he later delivered to 2/3 of participants.

Upon completion of the survey (for the thirty students, or 52%, who actually finished it), Survey Sam was either silent or provided one of two affective messages with a human voice: apologetic or empathetic. The script for the apologetic agent was based on the strategies in the Cross-Cultural Speech Act Realization Project [6] and the script for the empathetic agent paralleled the apologetic script based on Roger's [7] emotive-cognitive description of empathy. Table 1 lists the scripts used in this study.

Table 1. Scripts for Apologetic and Empathetic Messages

Message	**Scripts**
Apologetic	*"I'm really sorry that this problem happened to you. I know that the problem could have been avoided on our part, and it was not your fault. I promise that I will report this problem to the system administrator so that it will never happen again. Please take a few minutes to describe your experiences from the previous screens. Thank you."*
Empathetic	*"It must have been very frustrating trying to finish the survey with the problem you were experiencing. I sympathize with how you feel. I wish that I could have helped you to overcome this problem. Please take a few minutes to describe your experiences from the previous screens. Thank you."*

2.3 Post-survey

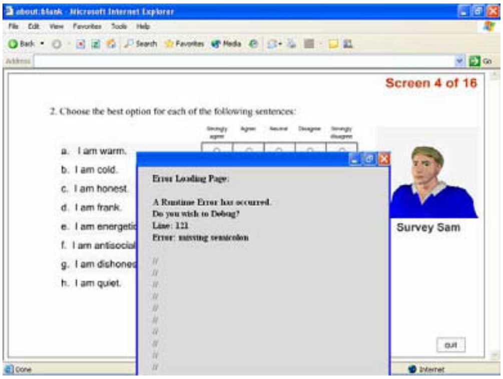
The post-survey assessed the dependent variables of agent competency, agent believability, survey enjoyment, survey frustration level, and attribution of the cause of the frustration. Agent competency and agent believability measures were adopted from API (Agent Persona Instrument) developed by Ryu and Baylor [13].

Three to five items were used to measure each dependent variable and each employed 5-item Likert scales. Internal consistency reliabilities (Cronbach's alpha) for Agent competency, Agent believability, Survey enjoyment, and Survey frustration level measure were .90, .74, .98, .88, and .75, respectively.

2.4 Procedure

A total of 289 emails were sent out to invite students to participate in a web-based personality survey and receive a free movie ticket upon completion. Respondents could complete the survey within the following four weeks.

The 55 participants who began the survey first provided demographic information and information regarding their computer self-efficacy. Following this, they completed items from the Big Five personality survey, as presented on five successive screens, with eight items per screen. Beginning on the second screen of the Big Five survey, a pop-up window appeared and covered up the survey items (see Figure 1). This pop-up window was designed to stimulate annoyance and frustration in the participants. The participants had to move the pop-up window out of the way in order to answer the survey questions (the window would not close by pressing the red "X" button). Because the pop-up window moved back to the original spot after five to nine seconds, participants had to repeatedly move the pop-up window out of the way to respond to the survey. Indeed, this was such a frustrating experience that only 30 of the initial 55 participants completed the survey.


about:blank - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address
Go
Screen 4 of 16
Strongly agree
Agree
Neutral
Disagree
Strongly disagree
Survey Sam
quit
Done
Internet

2. Choose the best option for each of the following sentences:

a. I am warm.
b. I am cold.
c. I am honest.
d. I am frank.
e. I am energetic
f. I am antisocial
g. I am dishonest
h. I am quiet.

Error Loading Page:

A Runtime Error has occurred.
Do you wish to Debug?
Line: 121
Error: missing semicolon

Figure 1. Screen shot with the pop-up window as an obstacle to answer survey questions

After completing the personality survey, the agent was either silent or provided an affective message (apologetic or empathetic). Next, students completed a post survey to assess agent competency, agent believability, survey enjoyment, survey frustration level, and attribution of the cause of the frustration.

2.5 Data analysis and design

A planned contrast with alpha level set at .05 was conducted to compare each dependent variable between those receiving no message (silent agent) and those receiving an affective message (either apologetic or empathetic). An independent sample t-test with alpha level at .05 was conducted to compare each dependent measure between the apologetic-message and empathetic-message groups. Students' perception of attribution of problem cause was analyzed with a one-way ANOVA, across the three agent conditions (silent, apologetic, empathetic).

3. Results

3.1. Impact of affective message

The major research question in this study was concerned with the effect of interface agent message (or absence). The descriptive statistics for each dependent variable are presented in Table 2.

Table 2. Means and standard deviations of dependent variables across groups.

	Agent competency		Agent Human-like		Agent belivability		Survey enjoyment		Survey frustration	
Message	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*
Apologetic (n=11)	3.18	.68	3.00	.51	3.00	.94	3.25	.96	3.24	.82
Empathetic (n=9)	2.78	.93	2.78	.83	3.89	.60	3.06	1.09	3.15	.94
Silent (no message) (n=10)	3.22	.47	2.85	.39	3.23	.39	3.33	.53	2.23	.93

For survey frustration, the result showed there was a statistically significant difference between those receiving an affective message and those receiving no agent message, $t(27) = 2.772$, p=.01, d=1.12, a large effect, indicating that students who received an agent message reported significantly higher frustration from taking the on-line survey than students who did not receive a message.

An independent sample t-test setting alpha level at .05 was conducted to compare each dependent variable between participants receiving an apologetic message and those receiving an empathetic message. Results revealed that for agent believability there was a statistically significant difference between the apologetic-message group and empathetic-message group, $t(18)= -2.445$, $p<.05$, d=1.16, a large effect, indicating that students in the empathetic-message group believed the animated agent more (e.g., believed that Survey Sam "meant what he said," and "was sincere in what he said") than students in the apologetic-message group.

3.2. Attribution of cause of problem

Students also rated their attribution of the cause of the problem. Descriptive statistics for the attribution of problem cause are presented in Table 3.

Table 3. Descriptive statistics for attribution of problem cause

Students attributed the problem malfunction to ...

	Themselves		Survey Sam		Computer software		Internet	
Message	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*
Apologetic (n=11)	1.82	.75	3.00	1.27	3.82	1.08	3.09	1.14
Empathetic (n=9)	1.56	.73	2.00	1.00	4.22	.83	2.67	1.32
Silent (no message) (n=10)	2.50	.53	2.90	.74	3.60	.70	3.30	.68

(Range of 1-5, where 1=SD and 5=SA)

A one-way ANOVA setting the alpha level at .05 was conducted to examine whether students attributed the cause of the problem to themselves, to Survey Sam, to the computer software, or to the Internet. The ANOVA yielded a significant overall difference, $F(2,29) = 5.03, p < .05\ \eta^2 = .27$. Follow-up Fisher's least significant difference (LSD) tests were performed to determine whether significant differences occurred between the mean scores for each pair of treatments. These tests revealed that those in the silent agent (no message) group tended to attribute the problem to themselves more than the other two message groups (p < .05). There was no statistically significant difference between the apologetic-message agent group and the empathetic-message agent group.

The ANOVA was also conducted to determine whether there were differences between groups in attributing the cause of the problem to Survey Sam. As expected, those receiving an apologetic message tended to attribute the problem to Survey Sam (p<.05). This validates the treatment, as it indicates that participants believed Survey Sam when he apologized and took responsibility for the problem.

4. Discussion

Results indicate that the presence of an affective message contributed to participants reporting significantly greater frustration. This indicates that they resonated with and believed the agent, as his message essentially "re-activated" their frustration, validating it and amplifying it. More importantly, students who received the agent's affective message also tended to attribute the cause of the frustration to the program (rather than themselves). Given that the problem was indeed out of their control, implementations like this that can reassure users that they are not at fault are of importance; indeed, this was only a five-sentence intervention, yet it yielded a large effect (over a standard deviation) Future research should consider the nature of this self-reported "frustration" and its relative weight in relation to users' attributions of the cause.

Results also indicated that students who received the empathetic message rated the agent as more believable than students in the apologetic-message group. Since the empathetic message conveyed an understanding of the participant's perspective rather than focusing on responsibility, it may have had the effect of making the participant feel that he/she and the agent were figuratively "in the same boat." This might have provided the participant with the perception of the interface agent as an understanding bystander instead of a responsible/apologetic or non-responsive helper. Also, the delay of the apology from the initial occurrence of the problem may have lessened the credibility of the apologetic message in terms of perceived sincerity. In addition, an apologetic message that conveys responsibility for the problem may also place the agent in an inferior position, i.e., the agent may be perceived as someone who has failed in avoiding technical problems. Either way, it is interesting that such a brief message from a non-human, computer-based, interface agent has such a profound impact, in line with findings by Reeves and Nass [14].

In retrospect, given that 25 of the 55 respondents who began the survey did not finish it (a 45.5% attrition rate), the survey was likely *too* frustrating. Another limitation is that the experiment had a low number of participants per condition (9, 10, and 11 respectively). However, in spite of the relatively low statistical power, the results were statistically significant with large effect sizes (d > 1.0). Another important consideration is that participants completed the study at their own computer and chosen time/place. While control in implementation was thus lost, ecological validity was enhanced, as this type of computer-based frustration could only be authentically simulated in a real context. Despite these limitations, the import of the findings is that the presence and nature of an affective message can impact how a user perceives frustration. These findings provide the catalyst for further research in the development of frustration-mitigating support for computer-based contexts.

Future research should include a control group to isolate the message(s) from the interface agent as the delivery mechanism. Future studies could also consider the timing of the message, including messages delivered during each problem occurrence rather than after-the-fact. Future studies could also track user interactions to determine when participants quit during a frustrating task and could compare participant personality characteristics with their frustration levels, attribution perceptions, and attitudes. *Also, note that we are in process of collecting more user data over the next weeks.*

5. Acknowledgments

This work was supported by the National Science Foundation, Grant IIS-0218692.

References

[1] N. Frijda, *The Emotions*: Cambridge University Press, 1986.

[2] R. S. Lazarus, *Emotion and adaption*. New York: Oxford University Press, 1991.

[3] R. Barker, "The Effect of Frustration upon the Cognitive Ability," *Character and Personality*, vol. 7, pp. 145-150, 1938.

[4] R. Bias, & Mayhew, D., "Cost-justifying usability." San Francisco: Morgan Kaufmann Publishers, 1994.

[5] B. W. Darby, & Schlenker, B. R., "Children's reactions to apologies," *Journal of Personality and Social Psychology*, vol. 43, pp. 742-753, 1982.

[6] S. Blum-Kulka, J. House & G. Kasper, "Cross-cultural pragmatics: Requests and apologies." Norwood, NJ: Ablex, 1989.

[7] C. R. Rogers, *A way of being*. Boston, MA: Houghton Mifflin Company, 1980.

[8] J. Mori, H. Prendinger, and M. Ishisuka, "Evaluation of an Embodied Conversational Agent with Affective Behavior," presented at AAMAS (Autonomous Agents - Multi Agent Systems) International Conference, Melbourne, Australia, 2003.

[9] N. Wang, W. L. Johnson, P. Rizzo, E. Shaw, and R. E. Mayer, "Experimental Evaluation of Polite Interaction Tactics for Pedagogical Agents," presented at International Conference on Intelligent User Interfaces, San Diego, CA, 2005.

[10] A. L. Baylor, E. Shen, D. Warren, and S. Park, "Supporting learners with math anxiety: The impact of pedagogical agent emotional and motivational support," presented at Workshop on "Social and Emotional Intelligence in Learning Environments," held at the International Conference on Intelligent Tutoring Systems., Maceió, Brazil, 2004.

[11] A. Bandura, *Self-efficacy: The exercise of control*. New York: W. H. Freeman., 1997.

[12] P. T. J. Costa and R. R. McCrae, "Revised NEO Personality Inventory (NEO PI-RTM) and NEO Five-Factor Inventory (NEO-FFI): Professional manual.," Psychological Assessment Resources., Odessa, FL 1992.

[13] J. Ryu and A. L. Baylor, "The Psychometric Structure of Pedagogical Agent Persona," *Technology, Instruction, Cognition & Learning (TICL)*, in press.

[14] B. Reeves and C. Nass, *The Media Equation*. Stanford, CA: CSLI Publications, 1996.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Computational methods for evaluating student and group learning histories in intelligent tutoring systems

Carole Beal [a,1], Paul Cohen [a]
[a] *USC Information Sciences Institute*

Abstract.
Intelligent tutoring systems customize the learning experiences of students. Because no two students have precisely the same learning history, traditional analytic techniques are not appropriate. This paper shows how to compare the learning histories of students and how to compare groups of students in different experimental conditions. A class of randomization tests is introduced and illustrated with data from the AnimalWatch ITS project for elementary school arithmetic.

Interacting with an intelligent tutoring system is like conversing with a car salesperson: No two conversations are the same, yet each goes in roughly the same direction: the salesperson establishes rapport, finds out what you want, sizes up your budget, and eventually makes, or doesn't make, a sale. Within and between dealerships, some salespeople are better than others. Customers also vary, for example, in their budget, how soon they intend to purchase, whether they have decided on a particular model, and so on. Of the customers who deal with a salesperson, some fraction actually purchase a car, so one can compare salespeople with a binomial tests or something similar. Indeed, any number of sound statistical comparisons can be drawn between the *outcomes* of dealing with salespeople: total revenues, distributions of revenues over car model classes, interactions between the probability of sale and model classes, and so on.

Similarly, one can evaluate intelligent tutoring systems on outcome variables: the number of problems solved correctly, or the fraction of students who pass a posttest, and so on. Consider the AnimalWatch tutoring system for arithmetic. Students between the ages of 10 and 12 worked on customized sequences of word problems about endangered species. They were provided with multimedia help when they made errors [1]. The word problems provided instruction in nine topics, including addition, subtraction, multiplication and division of integers, recognizing the numerator and denominator of a fraction, adding and subtracting like and unlike fractions and mixed numbers, and so on. Previous analyses focused on outcome measures such as topic mastery estimates maintained by the student model component of the AnimalWatch ITS. These analyses indicated that students who received rich multimedia help when they made errors (the Heuristic condition) had higher topic mastery scores than peers who worked with a text only version of the ITS which provided only simple text messages (e.g., "try again") [2].

[1] Correspondence to: Carole Beal, USC Information Sciences Institute Tel.: 310 448 8755; E-mail: cbeal@isi.edu.

Outcome variables can provide evidence of learning from an ITS. However, they tell us nothing about the individual student's experience while working with the tutor. Students might reach similar outcome points via quite different sequences of problems, or learning trajectories, some of which might be more effective, efficient or well-matched to particular students. Thus, if our interest is in the process of learning, then we should evaluate the efficacy and other attributes of sequences of problem-solving interactions. The challenge is that, by definition, each student's learning experience with an ITS is unique. For example, the AnimalWatch ITS includes more than 800 word problems, most of which can be customized in real time to individual students. Those who worked with AnimalWatch took unique paths through an extremely large problem space, and each step in their trajectories depended on their prior problem solving history [3].

One approach to evaluating student progress and performance while working with an ITS has been to examine the reduction in the number of errors across sequences of problems involving similar skills [4,5]. Unfortunately, the utility of this approach is often limited due to the lack of sufficient problems of the same type and difficulty that can be used to form meaningful sequences. A more serious problem is that the elements of interactions in a problem sequence are not independent; the next problem a student sees depends on his or her unique learning history. This means that we cannot treat the student's experience as a sample of independent and identically distributed problems, nor can we rely on traditional statistical methods (analysis of variance; regression) that treat it as such [6].

In this paper, we present alternative methods to compare the learning experiences of students, and experimental groups of students. We illustrate these methods with student problem solving data from the AnimalWatch project; however, they are general.

1. Comparing Experiences

The first step is to create a multidimensional representation of the student's experience as a sequence of dependent interactions. For instance, the student might attempt problem 1, fail, get a hint, fail again, get another hint, succeed, and then move onto problem 17, which the tutor judges to the best next problem, given the observed sequence of interactions. Let $S_i = x_1, x_2, \ldots, x_n$ be the sequence of interactions for student i. In general the set of interaction types is quite large; for instance, the AnimalWatch tutor includes 807 problems, each of which is instantiated with a variety of operands; and 47 distinct hint types. Interactions have attributes in addition to their type. They take time, they are more or less challenging to the student, they succeed or fail, and so on. In fact, interaction x_i is a vector of attributes like the one in Figure 1. This is the 5th problem seen by student `x32A4EE6`, it involves adding two integers, it is moderately difficult, it required 142 seconds and one hint to solve correctly, and so on. The experience of a student is represented by a sequence of structures like this one. While our examples all focus on information about problems (topic, difficulty, time), the approach can be generalized to other characterizations of students' experience, such as the frequency and content of hints. That is, we identity aspects of interaction with the ITS that we want to consider in an evaluation and represent these in the vector x_i.

Although the problem instance in Figure 1 is unique, it belongs to several *problem classes*; for instance, it belongs to the class of `ADD-INTEGERS` problems with

```
PROBLEM-ID: 675 , NUMBER: 5 , STUDENT: #<STUDENT x32A4EE6> ,
TOPIC: ADDINTEGERS ,  OP1: 8155 , OP2: 2937, DIFFICULTY: 4.33
NUMSKILLS: 2 , TIME-REQUIRED: 142 , NTHINTOPIC: 3 ,
HINTS: (<HINT x3646D76>)
```

Figure 1. A single problem instance presented to a student by AnimalWatch

`DIFFICULTY = 4.33`. Such *class attributes* define problem classes. Another example is the number of different math skills required to solve problems in the class. Other class attributes are derived from the problem instances in the class. An important derived attribute is *empirical difficulty*, which we define as the number of problems in a class answered incorrectly divided by the total number of attempted problems in that class. In Section 6 we will see that empirical difficulty often differs from a priori estimates by the ITS developers of the difficulty of problems.

Once we have created vectors to represent the elements of interest of the student's interaction with the ITS, we can compare students. We want to perform several kinds of analysis:

- Compare two students' experiences; for example, assess whether one student learns more quickly, or is exposed to a wider range of topics, than another.
- Form clusters of students who have similar experiences; for example, cluster students according to the rates at which they proceed through the curriculum, or according to the topics they find particularly difficult.
- Compare groups of students to see whether their experiences are independent of the grouping variables; for example, tutoring strategies are different if students have significantly different experiences under each strategy.

2. General Method

These kinds of analysis are made possible by the following method. We will assume that each problem instance x seen by a student is a member of exactly one problem class χ.

1. Re-code each student experience $S_i = x_1, x_2, \ldots x_n$ as a sequence of problem classes $\sigma_i = \chi_i, \chi_j, \ldots \chi_m$.
2. Derive one or more functions $\phi(\sigma_i, \sigma_j)$ to compare two problem class sequences (i.e., two students' experiences). Typically, ϕ returns a real-valued number.
3. Students may be grouped into empirical clusters by treating ϕ as a similarity measure. Groups of students (e.g., those in different experimental conditions) can be compared by testing the hypothesis that the variability of ϕ within groups equals the variability between groups.

Expanding on the last step, let G_i be a group comprising n_i sequences of problem classes (one sequence per student), so there are $C_i = (n_i^2 - n_i)/2$ pairwise comparisons of sequences. If we merge groups G_i and G_j, there are $C_{i \cup j} = ((n_i + n_j)^2 - (n_i + n_j))/2$ pairwise comparisons of all sequences.

Let

$$\delta(i) = \sum_{a,b \in G_i} \phi(a,b) \tag{1}$$

be the sum of all pairwise comparisons within group G_i. If groups G_i and G_j are not different, then one would expect

$$\Delta(i,j) = \frac{(\delta(i) + \delta(j))/(C_i + C_j)}{\delta(i \cup j)/C_{i \cup j}} = 1.0 \tag{2}$$

This equation generalizes to multiple groups in the obvious way: If there are no differences between the groups then the average comparison among elements in each group will equal the average comparison among elements of the union of all the groups.

3. Hypothesis Testing by Randomization

We introduce randomization testing for two groups, though it generalizes easily to multiple groups. In the previous section we introduced a test statistic $\Delta(i,j)$ and its expected value under a null hypothesis, but not its sampling distribution. The sampling distribution of a statistic under a null hypothesis H_0 is the distribution of values of the statistic if H_0 is true. Typically H_0 is a statement that two things are equal, for instance, $H_0 : \Delta(i,j) = 1$. If the test statistic has an improbable value according to the sampling distribution then H_0 probably is not true. We reject H_0 and report the probability of the test statistic given H_0 as a *p value*.

Suppose one has a statistic that compares two groups i and j, such as $\Delta(i,j)$ (Eq. 2). Under the null hypothesis that the groups are not different, an element of one group could be swapped for an element of the other without affecting the value of the statistic very much. Indeed, the elements of the groups could be thoroughly shuffled and re-distributed to *pseudosamples* i^* and j^* (ensuring that the pseudosamples have the same sizes as the original samples i and j) and the statistic could be recomputed for the pseudosamples. Repeating this process produces a distribution of *pseudostatistics* which serves as the sampling distribution against which to compare the test statistic.

Randomization is non-parametric, it makes no assumptions about the distributions from which samples are drawn; and it can be used to find sampling distributions for any statistic.

The hypothesis testing procedure for comparing two groups, i and j, of students, then, is to derive the test statistic $\Delta(i,j)$ as described earlier, then throw all the students into a single group, shuffle them, draw pseudosamples i^* and j^*, compute $\Delta^*(i^*,j^*)$ and increment a counter c if $\Delta^*(i^*,j^*) > \Delta(i,j)$. After repeating the process k times, the p value for rejecting the null hypothesis that the groups are equal is c/k.

3.1. About the Implementation

Comparing each student to every other is quadratic, repeating the process for each pseudosample adds a linear factor. Note also that the denominator of Eq. 2 is calculated only once; only the numerator changes when we draw pseudosamples. In practice, one can make the procedure run very fast by not actually drawing pseudosamples from the original sample but, rather, shuffling pointers into the original sample. This requires little more space than it takes to store the original samples and keeps the space complexity of the algorithm very low. The analyses in the examples below involve a few dozen students in each of two samples and 1000 pseudosamples, and none takes more than two minutes on a Macintosh G4.

4. Example: Comparing the progress of students in different conditions

Suppose we want to assess the distribution of topics encountered by a student after ten, twenty, ... problems, and compare students to see whether they progress through the topics in the same way. As noted earlier, AnimalWatch presented nine topics. Let $s_{i,t} = n_1, n_2, ...n_9$ represent the number of problems on each of nine topics encountered by student i at time t. Said differently, we imagine the progress of the student at time t as a point in nine-dimensional space. If we measure the progress of the student at regular intervals, we get a trajectory through nine-dimensional space. Two students may be compared by summing the Euclidean distances between corresponding points in this space:

$$\phi(\sigma_a, \sigma_b) = \sum_{t=0,10,20,...} \sqrt{\sum_{i=1,2,...9} (n_{i,a} - n_{i,b})^2} \tag{3}$$

We used the randomization method to compare progress for students in the Text and Heuristic experimental conditions, described earlier. We looked at each student after 10, 20, ..., 90 problems and recorded how many problems on each of nine topics the student solved. Students were compared with the function ϕ in Eq 3. The test statistic $\Delta(Text, Heuristic) = 0.981$ was rejected only twice in 1000 randomization trials, so we can reject the null hypothesis that progress through the nine-topic problem space is the same for students in the Text and Heuristic conditions, with $p = .002$.

It is one thing to test whether student in different experimental groups are different, another to visualize *how* they are different. In the previous example the trajectories are in a nine-dimensional space. However, the progress of each student through this space may be plotted as follows: Let $\mathcal{P}(s, t, c)$ be the proportion of problems in problem class c solved correctly by student s in the first t problems seen by that student. For instance, $\mathcal{P}$(1,30,addintegers)= .6 means that of the addintegers problems in the first 30 problems seen by student 1, 60 % were solved correctly. Let $\mathcal{N}(s, t, p)$ denote the number of problem classes for which $\mathcal{P}(s, t, c) > p$. For example, $\mathcal{N}(1, 30, .5) = 2$ means that in the first 30 problems, student 1 encountered two problem classes for which she solved 50% of the problems correctly. Let $V_{\mathcal{N}}(s, p) = [\mathcal{N}(s, 10, p), \mathcal{N}(s, 20, p), \mathcal{N}(s, 30, p)...]$, that is, the sequence of values of $\mathcal{N}$ for student s after $10, 20, 30...$ problems. Such a sequence represents progress for a student in the sense that it tells us how many classes of problems a student has solved to some criterion p after $10, 20, 30...$ problems.

To visualize the progress of a student one may simply plot $V_{\mathcal{N}}(s, p)$, and to compare groups of students one may plot the mean $V_{\mathcal{N}}(s, p)$ for students within groups. This is done in Figure 2. The vertical axis is mean $\mathcal{N}(s, t, p)$ averaged over students in a group, the horizontal axis is t, the number of problems attempted by the students. Here, t ranges from 10 to 100 problems. The higher of the two lines corresponds to the Heuristic condition, the lower to Text. One sees that on average, a student in the Heuristic condition masters roughly five topics to the criterion level of 50% in the first 100 problems, whereas students in the Text condition master only 3.5 topics to this level in the same number of attempts. These curves also can be compared with our randomization procedure, and are significantly different.

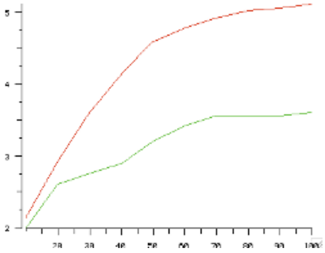


Figure 2. Mean number of problem classes mastered to the 50% criterion level as a function of the number of problems attempted by the students. Upper curve is Heuristic condition, lower is Text.

5. Example: Comparing the distribution of problems seen by students in different conditions

We will use data from the AnimalWatch project to illustrate the approach. Students were taught about nine arithmetic topics. Each student can therefore be represented as a vector of nine numbers, each representing the number of problems on a given topic that the student solved correctly, ordered on the basis of our empirical difficulty measure derived above (although the vector might represent other attributes of interest).

Let $\sigma_m(i)$ be the ith value in the vector for student m. Two students may be compared by

$$\phi(\sigma_m, \sigma_n) = \sum \text{abs}(\sigma_m(i) - \sigma_n(i)) \tag{4}$$

that is, the sum of the absolute differences in the numbers of problems solved correctly on each topic.

In this example, we will compare the learning experiences of students who worked with two different versions of the AnimalWatch ITS: Some students worked with a version that provided only minimal, text-based help in response to errors (Text). Other students worked with a version that provided students with rich, multimedia hints and explanations (Heuristic). Figure 3 shows the mean number of problems on each topic solved by students in the Text and Heuristic conditions, with 95% confidence intervals around the means. One might be tempted to run a two-way analysis of variance on these data with Topic and Condition as factors, but remember that the problems seen by a student are not independent, the tutor constructed a unique sequence of problems for each student, and the cell sizes are quite unequal, all of which violate assumptions of the analysis of variance. The randomization method makes no such assumptions. We compared the Text and Heuristic conditions with the randomization procedure described earlier. The test statistic $\Delta(Text, Heuristic) = .963$ was exceeded in every one of 1000 randomization trials, so we can reject the null hypothesis that the conditions are equal with $p < .001$. Thus, we conclude that, even though students had unique experiences with the ITS, those who received multimedia help in response to errors solved more problems correctly, across all topics, relative to students who received only limited, text-based help.

The total number problems solved by students was not the same in the Text and Heuristic conditions. This might account for the significant result. We can run the analysis differently, asking of each student what fraction of the problems she saw in each problem class she answered correctly. In this case we are comparing probabilities of correct responses, not raw numbers of correct responses. Repeating the randomization procedure with this new function for comparing students still yields a significant result, albeit less extreme: The test statistic $\Delta(Text, Heuristic) = .973$ was exceeded in 950 of 1000 trials, for a p value of 0.05.

By contrast, the p value for a comparison of girls and boys was 0.49, there is no reason to reject the null hypothesis that girls and boys correctly solved the same numbers of problems on all topics.

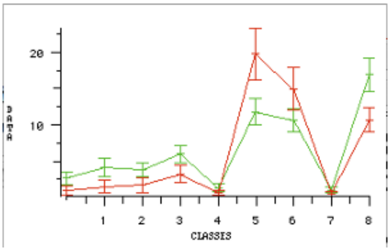


Figure 3. Mean correct number of problems for Heuristic and Text conditions.

6. Example: Change in Empirical Difficulty

As a final example of methods for comparing student experiences, we return to the idea of empirical difficulty, introduced in Section 1. We define the empirical difficulty of a problem as the number of unsuccessful attempts to solve it divided by the total number of attempts to solve it. Figure 4 shows the empirical difficulty of the nth problem for the Heuristic and Text groups. That is, the horizontal axis represents where a problem is encountered in a sequence of problems, the vertical axis represents the proportion of attempts to solve that problem which failed. Regression lines are shown for the Heuristic and Text groups. It appears that the empirical difficulty of problems in the Heuristic group is lower than that of the Text group, or, said differently, Heuristic students solved a higher proportion of problems they encountered. This appears to be true wherever the problems were encountered during the students' experience.

We can test this hypothesis easily by randomizing the group to which students belong to get a sampling distribution of mean empirical problem difficulty. This result is highly significant: In 1000 randomized pseudosamples the mean difference in problem difficulty between Heuristic and Text, 0.094, was never exceeded. One also can randomize the group to which students belong to get a p value for the difference between the slopes of the regression lines. This p value is .495, so there is no reason to reject the hypothesis that the regression lines have equal slope. In other words, the change in empirical problem difficulty as a function of when the problem is encountered, a slightly positive relationship, is the same for Heuristic and Text students.

In conclusion, we demonstrated that students' experiences with an ITS are sequences of multidimensional, dependent observations, and yet they are not beyond the reach of statistical analysis. We showed how to represent students' learning trajectories and how to test hypotheses about them with randomization methods.

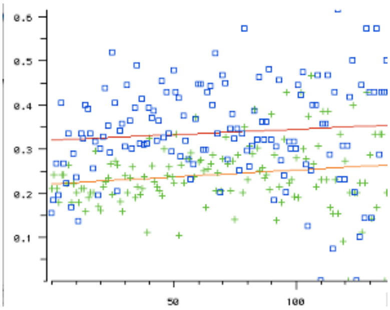


Figure 4. Empirical problem difficulty as a function of when problems are encountered.

Acknowledgments

We thank Dr. Ivon Arroyo for her work on constructing the AnimalWatch dataset. We also thank the students, parents, and staff of the schools that participated in the AnimalWatch project. The original AnimalWatch project was supported by National Science Foundation HRD 9714757. Preparation of this paper was supported by National Science Foundation REC 0411886 and HRD 0411532.

References

[1] Beal, C. R., & Arroyo, I. (2002). & The AnimalWatch project: Creating an intelligent computer mathematics tutor. In S. Calvert, A. Jordan, & R. Cocking (Eds.), Children in the digital age (pp. 183-198).

[2] Beck, J., Arroyo, I., Woolf, B., & Beal, C. R. (1999). An ablative evaluation. In Proceedings of the 9th International Conference on Artificial Intelligence, pp. 611-613, Paris: ISO Press.

[3] Beck, J. E., Woolf, B. P., & Beal, C. R. (2000). Learning to teach: A machine learning architecture for intelligent tutor construction. Proceedings of the Seventeenth National Conference on Artificial Intelligence, Austin TX.

[4] Arroyo, I. (2003). Quantitative evaluation of gender differences, cognitive development differences, and software effectiveness for an elementary mathematics intelligent tutoring system. Doctoral dissertation, University of Massachusetts at Amherst.

[5] Mitrovic, A., Martin, B., & Mayo, M. (2002). Using evaluation to shape ITS design: Results and experiences with SQL Tutor. Using Modeling and User Adapted Instruction, 12, 243-279.

[6] Cohen, P. R. (1995). Empirical methods for artificial intelligence. Cambridge MA: MIT Press.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Engagement tracing: using response times to model student disengagement

Joseph E. BECK
Center for Automated Learning and Discovery
Project LISTEN (www.cs.cmu.edu/~listen), Carnegie Mellon University
RI-NSH 4215, 5000 Forbes Avenue, Pittsburgh, PA. USA 15213-3890

Abstract. Time on task is an important predictor for how much students learn. However, students must be focused on their learning for the time invested to be productive. Unfortunately, students do not always try their hardest to solve problems presented by computer tutors. This paper explores student disengagement and proposes an approach, engagement tracing, for detecting whether a student is engaged in answering questions. This model is based on item response theory, and uses as input the difficulty of the question, how long the student took to respond, and whether the response was correct. From these data, the model determines the probability a student was actively engaged in trying to answer the question. The model has a reliability of 0.95, and its estimate of student engagement correlates at 0.25 with student gains on external tests. We demonstrate that simultaneously modeling student proficiency in the domain enables us to better model student engagement. Our model is sensitive enough to detect variations in student engagement within a single tutoring session. The novel aspect of this work is that it requires only data normally collected by a computer tutor, and the affective model is statistically validated against student performance on an external measure.

1. Introduction

Time on task is an important predictor for how much students learn. However, it is also important to ensure students are engaged in learning. If students are disinterested, learning will not be efficient.

Intelligent tutoring system (ITS) researchers sometimes have an implicit model of the student's engagement; such models help deal with the realities of students interacting with computer tutors. For example, the Reading Tutor [1] asks multiple-choice questions for the purpose of evaluating the efficacy of its teaching interventions. Unfortunately, if students are not taking the assessments seriously, it can be difficult to determine which intervention is actually most effective. If a student hastily responds to a question after just 0.5 seconds, then how he was taught is unlikely to have much impact on his response. Screening out hasty student responses, where students are presumably not taking the question seriously, has resulted in clearer differences between the effectiveness of teaching actions compared to using unfiltered data [2].

A different use of implicit models of student attitudes is the AnimalWatch mathematics tutor [3]. From observation, some students would attempt to get through problems with the minimum work necessary (an example of "gaming the system" [4]). The path of least resistance chosen by many students was to rapidly and repeatedly ask for more specific help until the tutor provided the answer. Setting a minimum threshold for time spent on the current problem, below which the tutor would not give help beyond "Try again" or "Check your work," did much to curtail this phenomenon.

In both the cases mentioned above, a somewhat crude model was added to an ITS to account for not all students being actively engaged: students who spent more time than the threshold were presumed to be trying, those who spent less time were presumed to be disengaged. These ad hoc approaches have drawbacks: differences among students and questions were ignored. Furthermore these approaches are unable to detect changes in student engagement over time in order to provide better tutoring.

This paper introduces a new technique, *engagement tracing*, to overcome these shortcomings. If the tutor can detect when students are disengaged with an activity it can then change tactics by perhaps asking fewer questions or at the very least disregarding the data for the purposes of estimating the efficacy of the tutor's actions.

2. Domain being modeled

This paper focuses on modeling disengagement by examining student performance on multiple-choice cloze questions [5]. The 2002-2003 Reading Tutor generated cloze questions by deleting a word (semi) randomly from the next sentence in the story the student was reading. The distractors were chosen to be words of similar frequency in English as the deleted word. The Reading Tutor read the sentence aloud (skipping over the deleted word) to the student and then read each response choice. The student's task was to select the word that had been deleted from the sentence. Since the process of generating cloze questions was random, it was uncommon to see repeats of questions and response choices, even when considering hundreds of students using the tutor. There are four types of cloze questions: sight, easy, hard, and defined. The cloze question's type is based on the word that was deleted; sight word questions were for very common words, hard questions were for rarer words, and defined word questions were for words a human annotated as probably requiring explanation. See [2] for additional details about how the cloze question intervention was instantiated in the Reading Tutor.

Which cloze data are relevant? One concern was whether students would take cloze questions seriously. Project LISTEN member Joe Valeri suggested that if students weren't really trying to get the question correct, they would probably respond very quickly. As seen in Figure 1, student performance on cloze questions was strongly related to how much time they spent answering a question. Since chance performance is 25% correct, it is safe to infer that students who only spent one second before responding were not trying to answer the question and were probably disengaged. Similarly, a student who spent 7 seconds was probably engaged. But what of a student who spent 3 seconds? Students responding after 3 seconds were correct 59% of the time, much better than baseline of 25% but not nearly as high as the 75% correct attained by students who spent 5 seconds. Should we consider such a response time as a sign of disengagement or not?

We consider four general regions in Figure 1. In region R1, students perform at chance. In region R2, student performance is improving as more time as spent. In region R3, performance has hit a plateau. In region R4, performance is gradually declining as student spend more time before responding to the question.

Although there is certainly a correlation between student performance and student engagement, we do not treat the decline in student performance in region R4 as a sign of disengagement. Without more extensive instrumentation, such as human observers, we cannot be sure why performance decreases. However, it is likely that students who know the answer to a question respond relatively quickly (in 4 to 7 seconds). Students who are less sure of the answer, or who have to answer on the basis of eliminating some of the choices based on syntactic constraints, would take longer to respond. This delay is not a sign of disengagement; therefore, to maintain construct validity, we do not consider long response times to be a sign of disengagement. For purposes of building a model to predict the probability a student is disengaged, we only consider data in regions R1, R2, and R3.

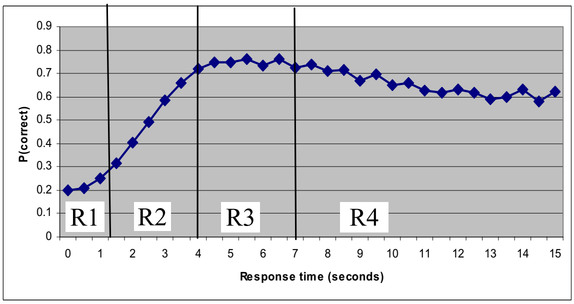


Figure 1. Student proportion correct on cloze questions plotted by response time

Describing the relation between response time and performance. Throughout regions R1, R2, and R3, performance with respect to time is similar to a logistic curve. Therefore, we use item response theory [6] as a starting point for our modeling. Item response theory (IRT) provides a framework for predicting the probability a student with a particular proficiency will answer a question correctly.

Three parameter IRT models [6] are of the form $p(correct \mid \theta) = c + \frac{1-c}{1+e^{-a(\theta-b)}}$. In this equation, θ represents the student's proficiency. The other three parameters control the shape of the logistic curve: a is the discrimination parameter, and determines the steepness of the logistic curve; b is the item difficulty parameter, and controls how far left or right the curve is shifted, and c is the "guessing" parameter and provides a lower bound for the curve. Since our items are multiple choice questions with four responses, we set c to be 0.25.

For our work, we need to modify the standard IRT formula in three ways. First, rather than taking student proficiency as input, our model uses response time as an input. Second, we cannot estimate item parameters for every cloze question, as a pure IRT model would, since most questions were only seen once. Therefore, we estimate discrimination and item difficulty parameters for each of the four types of cloze questions. Since the difficulty parameter cannot capture the differences between questions of a particular type, we also include the length of the cloze question and response choices (as the number of characters). Longer questions are probably harder than shorter ones, and at the very least should take more time to answer. Finally, in IRT models, as students become more proficient the chances of a correct response increase to 100%. For our model, the upper bound on performance is considerably less than 100%. If a student does not know the answer, giving him additional time (unless he has resources such as a dictionary to help him) is unlikely to be helpful. Therefore we introduce an additional parameter, u, to account for the upper bound on student performance.

The form of our modified model is $p(correct \mid rt, L_1, L_2) = c + \frac{u-c}{1+e^{-a(-rt+b(L_1+L_2))}}$. Parameters a, b, and c have the same meaning as in the IRT model. The u parameter represents the upper bound on performance, and L_1 and L_2 are the number of characters in the question and in all of the response choices combined, respectively. The u parameter is equal to the maximum performance (found by binning response times at a grain size of 0.5 seconds, and selecting the highest average percent correct).

We estimate the a (discrimination) and b (difficulty) parameters separately for each type of cloze question using SPSS's non-linear regression function. All question types have a similar difficulty parameter; the difference in difficulty of the questions is largely accounted for by the longer question and prompts for more difficult question types. For predicting whether a student would answer a cloze question correctly, this model accounts for 5.1% of the variance for defined word questions, 12.3% for hard words, 14.5% for easy words, and 14.3% for sight words. These results are for testing and training on the same data set. However, the regression model is fitting only two free parameters (*a* and *b*) for each question type, and there are 1080 to 3703 questions per question type. Given the ratio of training data to free parameters, the risk of overfitting is slight, and these results should be representative of performance on an unseen test set.

Determining student engagement. Although our model can estimate the probability of a correct response given a specific response time, this model is not sufficient to detect disengagement. To enable us to make this calculation, we assume that students have two methods of generating responses:

1. If the student is disengaged, then he guesses blindly with a probability *c* of being correct.

2. If the student is engaged, then he attempts to answer the question with a probability *u* of being correct.

Given these assumptions, we can compute the probability a student is disengaged in answering a question as $\frac{u - p(correct \mid rt, L_1, L_2\ )}{u - c}$. For example, consider Figure 1; if a student took 3 seconds to respond to a question he had a 59% chance of being correct. The lower bound, *c*, is fixed at 25%. The upper bound, *u*, is the best performance in region R3, in this case 76%. So the probability the student is disengaged is (76% - 59%) / (76% - 25%) = 33%, and therefore a 67% chance that he is engaged in trying to answer the question.

This model form is similar to knowledge tracing [7], in that both are two-state probabilistic models attempting to estimate an underlying student property from noisy observations. Since this model concerns student engagement rather than knowledge, we call it *engagement tracing*.

To illustrate the above process, Figure 2 shows our model's predictions and students' actual performance on hard word cloze questions. To determine the student's actual performance, we discretize the response time into bins of 0.5 seconds and took the mean proportion correct within the bin. To determine the performance predicted by the model, we use the estimates for the *a, b,* and *u* parameters, and assume all questions are of the mean length for hard question types (47.8 character prompt + 26.3 character response choices = 74.1 characters). As indicated by the graph, students' actual (aggregate) performance is very similar to that predicted by the model; the r^2 for the model on the aggregate data is 0.954, indicating that the model form is appropriate for these data.

However, this model does not account for individual differences in students. For example, a very fast reader may be able to read the question and response choices, and consistently give correct answers after only 1.5 seconds. Is it fair to assert that this student is not engaged in answering the question simply because he reads faster than his peers? Therefore, to better model student engagement, we add parameters to account for the variability in student proficiency.

Accounting for individual differences. One approach to building a model to account for inter-student variability is to simply estimate the *a, b,* and *u* parameters for each student for each question type (12 total parameters). Unfortunately, we do not have enough data for each student to perform this procedure. Students saw a mean of 33.5 and a median of 22 cloze questions in which they responded in less than 7 seconds. Therefore, we first estimate the parameters for each question type (as described above), and then estimate two additional parameters for each student that apply across all question types. The new model form becomes $p(correct \mid rt, L_1, L_2) = c + \frac{accuracy(1-u) + u - c}{1 + e^{-a(-rt + speed * b(L_1 + L_2)}}$ where *accuracy* and *speed* are the student-specific parameters. The first additional parameter, *speed*, accounts for differences in the student's reading speed by adjusting the impact of the length of the question and response choices. The second parameter, *accuracy*, is the student's level of knowledge. Students who know more words, or who are better at eliminating distractors from the response choices will have higher asymptotic performance.

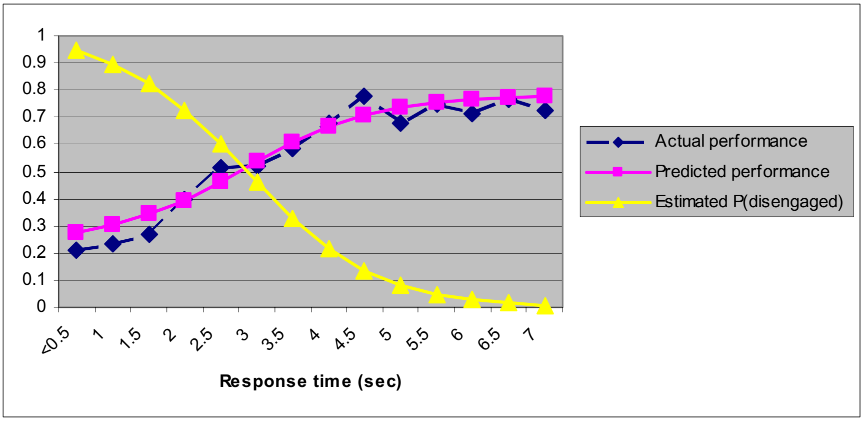


Figure 2. Empirical and predicted student behavior for hard word cloze questions

We estimate the student parameters with SPSS's non-linear regression procedure. The student-specific parameters are bounded to stop semantically nonsensical results. The *speed* parameter is forced to be in the range [0.33, 3] (i.e. it can model that students are three times faster or slower at reading than average) and the *accuracy* parameter is in the range [-2, 1] (i.e. students can not have performance over 100%). Thus we avoid obtaining a good model fit by assigning a student parameter a value that is implausible (such as reading 25 times faster than average).

3. Psychometric properties of model

There are two major psychometric properties: reliability, whether the measure is consistent, and validity, whether the model measures what it is supposed to measure. In our experimental design, for each cloze question a student encountered, we use our engagement tracing model to estimate the probability a student is engaged in answering the question. For each student, we take the mean probability of disengagement across all of the questions as a measure of the student's overall disengagement with the tutor.

Although our model's parameters are estimated from questions where students respond in fewer than 7 seconds, to estimate overall disengagement we use data from all cloze questions, even those with longer response times. Our belief is that students taking longer than 7 seconds to respond are engaged. As seen in Table 2, as response time increases the estimated probability of disengagement decreases, so including longer response times led the model to believe students were more engaged.

Students saw a mean of 88.7 and a median of 69 cloze questions. The mean probability of disengagement (for the student-specific model) is 0.093 and the median is 0.041. The probability of disengagement is positively skewed, with one student having a value of 0.671. This student saw 171 cloze items, so the high average disengagement is not a statistical fluke from seeing few items. Four students had disengagement scores over 0.5.

Reliability. To determine whether our engagement measure is psychometrically reliable, we use a split-halves approach by ordering each student's cloze data by time and assigning each cloze item to alternating groups (i.e. observation #1 is in group A, observation 2 is in group B, observation 3 is in group A, …). For each student, we then estimate the overall disengagement for A and for B. The corrected-split halves reliability is 0.95, comparable to the best psychometric instruments. Thus, our measure of disengagement is highly reliable.

Validity. To measure validity, we relate our measure of disengagement to existing tests of student performance and interest in the domain. Our hypothesis is that a measure of student disengagement should correlate negatively with student gains in reading over the course of the year. This hypothesis came from [4] as well as the intuition that an active, engaged learner is likely to make more progress than one who takes less initiative. We measure reading gains as the difference between the student's pretest and posttest on the (human-administered and scored) Woodcock Reading Mastery Test's [8] Total Reading Composite (TRC) subtest. We also examine how our measure of engagement correlates with the student's attitude towards reading as measured by the Elementary Reading Attitude Survey (ERAS) recreational reading subscale [9]. We have data and test scores for 231 students who were in grades one through six (approximately five- through twelve-year olds) during the 2002-2003 school year.

We compare three models of student engagement: a model with student-specific parameters (*speed* and *accuracy*), a model without the two student-specific parameters, and the percentage of questions to which a student responds to in less than 2.5 seconds, which corresponds to a ≈50% chance of engagement. Table 1 shows how the measures of disengagement, student attitude towards reading, and learning gains interrelate. These partial correlations hold constant student TRC pretest scores and student gender. All of the disengagement measures correlate with student gains in TRC at p<0.05, with the per-student model producing the strongest results. All correlations are in the intuitive direction: disengaged students have smaller learning gains while students with a positive attitude towards reading have higher gains.

Table 1. Partial correlations between disengagement, learning gains and reading attitude

	Measures of disengagement			Reading attitude
	Per-student model	Basic model	Response < 2.5 s	ERAS
TRC gain	-0.25 (p<0.001)	-0.16 (p=0.013)	-0.15 (p=0.023)	0.18 (p=0.007)
ERAS	-0.03	0.04	0.03	-

Somewhat surprisingly, none of the measures correlate with the student's attitude towards reading. Perhaps the measures of disengagement are unrelated to the student's overall attitude, but instead measure the student's specific feelings about working with the Reading Tutor or with its multiple choice questions.

4. Temporal properties of model

Although engagement tracing is psychometrically reliable, that does not mean student engagement is stable across time. We investigate two ways in which engagement can vary. Systematic change refers to students becoming consistently more or less engaged over the course of the year. Ephemeral change investigates whether our approach is sensitive enough to detect waxing and waning student engagement. For both investigations we focus on when cloze questions occur.

Systematic properties. To find systematic trends in student engagement, for each cloze question we compute how long the student has been using the Reading Tutor before encountering the cloze question, and then bin questions based on how many months the student has been using the tutor. During the first month, students have a mean disengagement of 6%. For each successive month the amount of disengagement increases until reach a plateau at the 4th month: 10.3%, 10.9%, 16.5%, 15.3%, and finally 16.5% during the 6th month of usage. Whether this result means students are becoming less engaged with the Reading Tutor or just bored with the questions is unclear.

Ephemeral properties. Presumably, student engagement should be similar across a small time interval, and vary more widely over a larger window. Can engagement tracing detect such transient effects? To answer this question, for a cloze question Q1, we pair Q1 with every successive cloze question seen by that student and compute the amount of intervening time between the questions. We then examine two models: the first correlates student engagement on Q1 and Q2; the second model computes a partial correlation between Q1 and Q2, holding constant the student's average level of disengagement throughout the year. Table 2 shows the results of this procedure.

Table 2. Detecting ephemeral properties of disengagement

Time between Q1 and Q2	Overall correlation	Partial correlation
< 1 minute	0.69	0.45
1 to 5 minutes	0.66	0.35
Later that day	0.63	0.21
Later that week	0.67	0.15
More than a week later	0.53	0.00

Overall, student performance on Q1 is strongly correlated with later performance on Q2. This result is not surprising, since a student presumably has an underlying level of engagement; thus we expect a strong autocorrelation. The partial correlation shows ephemeral trends in engagement. Specifically, student engagement on one question accounts for 19.8% of the variance in each measurement of engagement within a one-minute window, *even after controlling for the student's overall level of engagement throughout the year.* In contrast, a particular question only accounts for 2.3% of the variance of each measurement of student engagement later that week. This result both points to temporal trends in students using the Reading Tutor: engagement is much more consistent within a one- or five-minute interval than across successive days, and to the ability of engagement tracing to detect such differences.

5. Contributions, conclusions, and future work

Although by focusing on a single type of affect, namely disengagement, this work is narrower in scope than most prior work (e.g. [10-12]), it differs from that work by providing an empirical evaluation of whether the affective model relates to externally meaningful measures of real students. Also, the approach described in this paper does not require humans to rate user interactions (as in [12]) or measurements with biological sensors (as in [11]).

We have presented a means for analyzing the response times and correctness of the student responses to model overall level of engagement while using a computer tutor. This result is general as both response time and correctness are easily measurable by an ITS, do not require investing in new equipment, and are common across a wide variety of computer tutors.

The psychometric properties of the model include very strong reliability, and external validity to the extent of a moderate correlation with paper test scores. The model is sensitive enough to detect temporal changes in the student's level of engagement within a single session of using the tutor.

Future work with engagement tracing includes adding a temporal component to the model. Currently we simply take the mean of all student observations. Given the temporal nature of engagement, some means of discounting older observations is needed. To compare with knowledge tracing [7], this paper develops a framework comparable to the performance parameters (slip and guess), but does not yet have an equivalent to the learning parameters to account for initial student state and transitions between states.

This paper demonstrates that simultaneously modeling the student's proficiency and engagement allows us to better estimate his level of engagement than a model that ignores individual differences in proficiency. In the short-term, modeling a student's level of engagement enables predictions about how much an individual student will benefit from using a computer tutor. In the longer term, adapting the tutor's interactions to keep the learner happy and engaged—while not sacrificing pedagogy—is a fascinating problem.

Acknowledgements

This work was supported by the National Science Foundation, under ITR/IERI Grant No. REC-0326153. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation or the official policies, either expressed or implied, of the sponsors or of the United States Government. The author also thanks Ryan Baker and Cecily Heiner for providing useful comments on this work, and Jack Mostow for coining the term "engagement tracing" to describe the work.

References

1. Mostow, J. and G. Aist, *Evaluating tutors that listen: An overview of Project LISTEN*, in *Smart Machines in Education*, K. Forbus and P. Feltovich, Editors. 2001, MIT/AAAI Press: Menlo Park, CA. p. 169-234.
2. Mostow, J., J. Beck, J. Bey, A. Cuneo, J. Sison, B. Tobin, and J. Valeri, *Using automated questions to assess reading comprehension, vocabulary, and effects of tutorial interventions.* Technology, Instruction, Cognition and Learning, to appear. **2**.
3. Woolf, B.P., J.E. Beck, C. Eliot, and M.K. Stern, *Growth and Maturity of Intelligent Tutoring Systems: A Status Report*, in *Smart Machines in Education: The coming revolution in educational technology*, K. Forbus and P. Feltovich, Editors. 2001, AAAI Press. p. 99-144.
4. Baker, R.S., A.T. Corbett, K.R. Koedinger, and A.Z. Wagner. *Off-Task Behavior in the Cognitive Tutor Classroom: When Students "Game The System."* in *ACM CHI.* 2004.p. 383-390.
5. Entin, E.B., *Using the cloze procedure to assess program reading comprehension.* SIGCSE Bulletin, 1984. **16**(1): p. 448.
6. Embretson, S.E. and S.P. Reise, *Item Response Theory for Psychologists*. Multivariate Applications, ed. L.L. Harlow. 2000, Mahwah: Lawrence Erlbaum Associates. 371.
7. Corbett, A.T. and J.R. Anderson, *Knowledge tracing: Modeling the acquisition of procedural knowledge.* User Modeling and User-Adapted Interaction, 1995. **4**: p. 253-278.
8. Woodcock, R.W., *Woodcock Reading Mastery Tests - Revised (WRMT-R/NU).* 1998, Circle Pines, Minnesota: American Guidance Service.
9. Kush, J.C. and M.W. Watkins, *Long-term stability of children's attitudes toward reading.* The Journal of Educational Research, 1996. **89**: p. 315-319.
10. Kopecek, I., *Constructing Personality Model from Observed Communication*. 2003: Proceedings of Workshop on Assessing and Adaptive to User Atitudes and Effect: Why, When, and How? at the Ninth International Conference on User Modeling. p. 28-31.
11. Conati, C., *Probabilistic Assessment of User's Emotions in Educational Games.* Journal of Applied Artificial Intelligence, 2002. **16**(7-8): p. 555-575.
12. Vicente, A.d. and H. Pain. *Informing the Detection of the Students' Motivational State: an Empirical Study*. in *Sixth International Conference on Intelligent Tutoring Systems*. 2002.p. 933-943 Biarritz, France.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Interactive Authoring Support for Adaptive Educational Systems

Peter Brusilovsky, Sergey Sosnovsky, Michael Yudelson, Girish Chavan
School of Information Sciences, University of Pittsburgh
135 North Bellefield Ave. Pittsburgh, PA 15260, USA
{peterb, sas15, mvy3} @pitt.edu, chavang@upmc.edu

Abstract. A well-known challenge of adaptive educational systems is the need to develop intelligent content, which is very time and expertise consuming. In traditional approaches a teacher is kept at a distance from intelligent authoring. This paper advocates the involvement of teachers in creating *intelligent content*. We are presenting an approach to the development of intelligent content as well as an authoring tool for teachers that support our approach. This approach has two main stages: elicitation of concepts from content elements and the identification of a prerequisite/outcome structure for the course. The resulting sequence of adaptive activities reflects the author's view of the course's organization. The developed tool facilitates concept elicitation in two ways: it provides an author with an automatic indexing component and also allows her/him to edit the index using the domain ontology as an authoring map.

Introduction

An increasing number of adaptive and intelligent web-based educational systems [1] are reaching the point where they can be used in the context of a real classroom or online school, an area that up to now has been almost exclusively served by traditional non-intelligent and non-adaptive web-based educational systems [2]. Thanks to years of research, a multiple set of problems: representing the domain model, the procedural expertise, the knowledge about a student, as well as developing the interface can now be solved in a number of domains by relatively small research teams. The choice of the Web as implementation platform can help a small team solve problems of delivery, installation, and maintenance, thus making their intelligent systems available to hundreds and thousands of students. Yet, there is one "last barrier." Traditional static, non-intelligent web-based educational (WBE) systems and courses have provided something that almost no intelligent system developed by a small research team can offer – large amounts of diverse educational material. A high-quality traditional WBE course may have thousands of presentation pages, and hundreds of other fragments of learning material, including examples, explanations, animations, and objective questions created by a team of developers. In comparison, the number of presentation items in even the best intelligent WBE systems is well under one hundred and the number of other fragments of learning material, such as problems or questions, is no more than a few dozen. These numbers are certainly sufficient for a serious classroom study of the system, but still quite far from the resources needed for a practical web-based educational approach, namely one, which could support reasonable fragments of practical courses that are taught to large numbers of students, semester by semester.

The origin of this bottleneck is the established design paradigm of existing adaptive and intelligent educational systems. With this approach, a system is created by a team of expert developers and shipped to their users (teachers and students) as a whole. Within this approach, little can be done to magnify the volume of available educational content. We think that the move of adaptive and intelligent web-based educational systems (AIWBES) from labs to regular classrooms has to be supported by a change in the design paradigm. A significant increase of the role of teachers as principal users of intelligent educational systems must be supported by a parallel increase in their participation in the authoring process. We argue that the new paradigm would make teachers more active players in the authoring process by separating the authoring process into two parts: core AIWBES authoring and educational content authoring. Core authoring should comprise the development of the core functionality of AIWBES: knowledge representation, algorithms, interfaces, and core educational content. This part is not different from traditional authoring and should remain in the hands of a professional development team (which hopefully will include some prize-winning teachers). At the same time, the core of AIWBES should be designed in such a way as to allow the majority of the educational content (such as explanations, examples, problems) to be authored by teachers working independently of the development team (and possibly continuing long after the system is originally deployed).

The idea of involving teachers as content authors comes naturally to the developers of the practical AIWBES that are used in dozens of classrooms. It is not surprising that the first implementation of this idea by Ritter et al. [3] was done in the context of the PACT Algebra Tutor, the first AIWBES to make a leap from the lab to hundreds of classrooms [4]. Later, this idea was also explored in the context of AnimalWatch [5], another practical algebra tutoring system. This solution looks like it may be a silver bullet. Not only does it solve the "lack of content" bottleneck, but it also offers multiple additional benefits. The ability to contribute their favorite content transforms teachers from passive users of new technology into active co-authors. It turns an AIWBES which competes with the teacher into a powerful tool in the teacher's hands. A strong feature of traditional non-adaptive web-based educational systems is that while offering a core framework for web-based education, they also allow every teacher to author easily their own educational content. An AIWBES that allows teachers to add their own content will have a much better chance to compete with the non-intelligent systems which now dominate the educational arena.

The goal of this project is to investigate the use of teachers to develop educational content in a specific domain for AIWBES. The next section discusses the problems faced when supporting teachers as authors of intelligent content. The following sections explain how we address some of these challenges in an authoring system that creates advanced content in AIWBES for an introductory programming class. At the end, we summarize our results and discuss future work.

1. Supporting teachers as authors of adaptive and intelligent content

The teacher's involvement in the process of AIWBES authoring is recognized as both a need and as a research stream in AIWBES community. However, the original goal was also to involve teachers in the *core design* process. This direction of work brought little practical success. After a number of attempts to turn teachers into key developers of AIWBES, no one has the illusion that a teacher can design an AIWBES, even with the help of advanced authoring tools. As pointed out by Murray in his comprehensive overview of ITS authoring tools [6]: "The average teacher should not be expected to design ITSs any more than the average teacher should be expected to author a textbook in their field".

The new design paradigm offers teachers a different place in the process of AIWBES authoring. It leaves the core authoring in the hands of well-prepared design teams and gives teachers a chance to extend the system and fine tune it to their local needs by adjusting and adding to the educational content. Such division of labor is quite natural. Indeed, while it is rare for teachers to be able to create a textbook for their courses, many of them augment existing textbooks with their own examples, problems, questions, and even additional explanations of complicated concepts.

Still, the development of the content authoring tools for an AIWBES that can be used by regular teachers is a research problem that should not be underestimated. Teachers are much less prepared to handle the authoring than professional AIWBES developers, and require a significant level of support. The pioneering paper [3] provides a good analysis of problems and a set of design principles developed for solving the authoring problems that exist for a cognitive rule-based tutoring system.

The main issue here is that the content to be created for an AIWBES is really *intelligent content*. The power of intelligent content is in the knowledge behind its every fragment. Even the simplest presentation fragments of external content should be connected to the proper elements of domain knowledge (concepts) so that an AIWBES can understand what it is about, when it is reasonable to present it, and when it is premature. More complicated types of content, such as examples and problems, require that even more knowledge be represented, in order to enable an AIWBES to run the example or to support the student while he/she is solving a problem.

For example, adaptive educational hypermedia systems such as InterBook [7], AHA! [8], or KBS-Hyperbook [9] require every hypermedia page to be connected to a domain model concept in order for the server to know when to present them in an adaptive manner. Moreover, InterBook and AHA! require separating connected concepts from page prerequisites (concepts to know before reading a page) and page outcomes (concepts presented in the page). This knowledge has to be provided during the authoring process. As we have found during our work with InterBook, content authors have problems identifying concepts associated with content pages even if the number of concepts in the domain model is under 50. For adaptive hypermedia authoring this “concept indexing” becomes a major bottleneck. While a few pioneer systems such as KBS-Hyperbook [9] and SIGUE [10] allowed teachers to add additional content by indexing content pages with domain concepts, they provide no special support for teachers in the process of indexing. The AHA! System shows some progress towards this goal by providing a graphical authoring tool that will show connections between concepts and pages, but this tool becomes difficult to use when the number of concepts and pages approaches the level of that used in a practical classroom.

Traditionally, there are two ways to support humans in performing complicated tasks: an AI approach (i.e., make an intelligent system that will do this task for the user) and an HCI approach (i.e., provide a better interface for the humans to accomplish the task). In the case of indexing, it means that one must either develop an intelligent system that can extract concepts from a fragment of content or develop a powerful interface that can help the teacher do this manually. While both approaches are feasible, our team was most interested in a hybrid approach – a “cooperative” intelligent authoring system for the teachers that split the work between a human author and an intelligent tool so that both "agents" were able to “cooperate.” doing their share of work. We have started to explore this idea by developing a cooperative authoring system for the domain of programming. The goal of this system is to allow authors to collaboratively index interactive educational content (such as program examples or exercises) with domain model concepts while separating them into prerequisite and outcome concepts.

The following two sections describe our indexing approach and the system that implements it. These sections present two main stages of the approach: concept elicitation and prerequisite/outcome identification. In the first stage, a cooperative indexing tool extracts concepts from the content elements (examples, questions, presentation pages), grouped by the type of activity (i.e., all examples form one pool while all quizzes belong to another pool). In the second stage, a teacher-driven prerequisite/outcome identification algorithm separates the concepts connected with each content item into prerequisites and outcomes as required by the adaptive hypermedia system. While the cooperative authoring process has been used with two kinds of educational content, the following sections focus on one of these kinds – parameterized quizzes served by the QuizPACK system [11].

2. Content Indexing

There are no universally accepted recommendations as to which level is best to use when defining a concept in the computer programming domains. Some authors theorize that it has to be done on the level of programming patterns or plans [12]. Others believe that the main concepts should be related to the elementary operators [13]. According to the first point of view, the notion of pattern is closer to the real goal of studying programming, since patterns are what programmers really use. However, the second way is more straightforward and makes the burden of indexing more feasible. With the notable exception of ELM-PE [14], all adaptive sequencing systems known to us work with operator-level concepts. Our web-based cooperative indexing system allows us to combine two kinds of indexing. Simple operator-level indexing is performed by an automatic concept extractor, while more complicated higher-level indexing is performed by the author, using a graphical ontology-based tool.

Figure 1 demonstrates the interface for authoring QuizPACK parameterized questions. The main window is divided into two parts. The left part contains functionality for editing the text and different parameters of the question (details are not important for the topic of this paper). The right part facilitates the elicitation of the concepts used in the question. It provides an author with non-exclusive possibilities: to extract concepts automatically and/or to use a visual indexing interface based on the visualized ontology of available concepts. The following subsections discuss both modes.

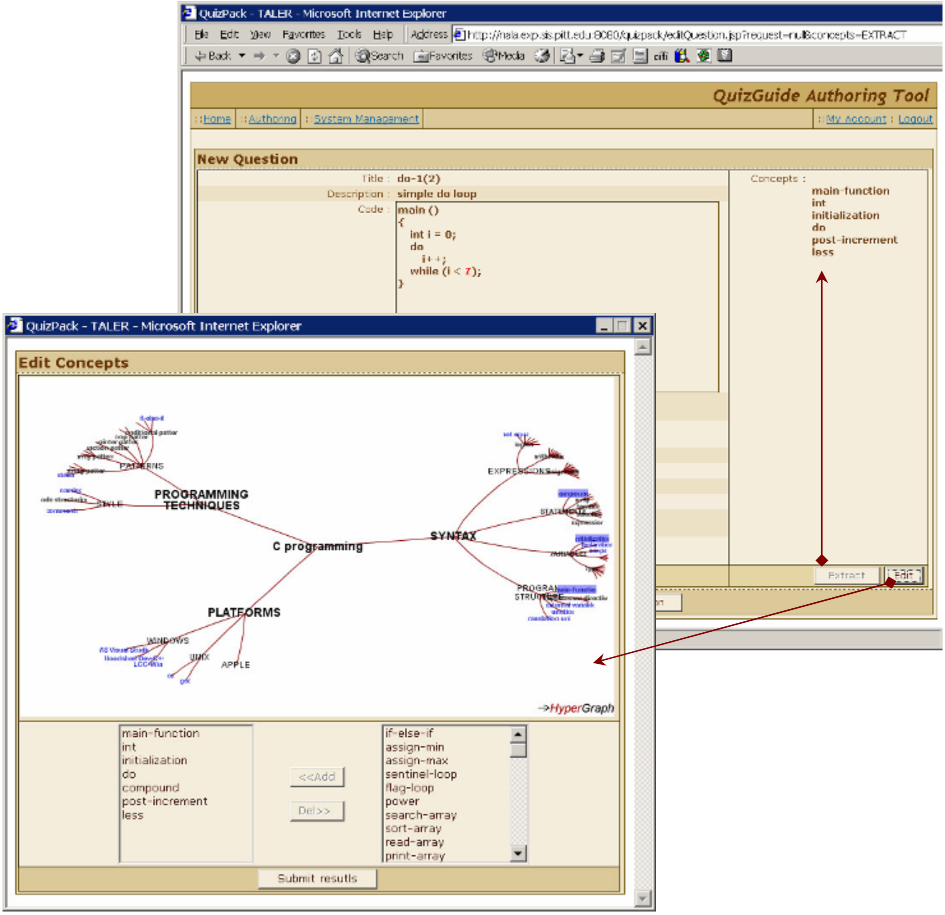
2.1. Automated Concept Extraction

Traditionally, the automatic extraction of grammatically meaningful structures from textual content and the determination of concepts on that basis is a task for the special class of programs called parsers. In our case, we have developed the parsing component with the help of two well-known UNIX utilities: lex and yacc. This component processes the source code of a C program and generates a list of concepts used in the program. Currently, about 80 concepts can be identified by the parser. Each language structure in the parsed content is indexed by one or more concepts, depending upon the amount of knowledge students need to have learned in order to understand the structure. For instance, the list of concepts in the right part of Figure 1 has been generated by the parser for the program code of the question in the left part of the figure. It is necessary to mention that each concept in this list represents not simply a keyword, found in the code, but a grammatically complete programming structure.

To launch the automatic indexing, an author clicks on the button *Extract* under the *Concepts* section of the interface. The list is then populated and the button dims out. If the code of a question has been changed, the button regains its clickability. This is done to prevent the author from losing the results of manual indexing, described in the next subsection.

2.2. Ontology as a Tool for Authoring Support

Automated indexing is not always feasible. Some higher order concepts involve understanding programming semantics that might be hard to extract. In more advanced courses like *Data Structure* or *Algorithm Design,* pattern-oriented questions may be popular. For example, there are several modifications of the sentinel loop. The parser we developed easily breaks such fragments of code into syntax concepts (which must be learned in order to understand the code), however, it is not reasonable to make it follow each and every possible configuration of the sentinel loop. We also should take into account that an author of content might not fully agree with the results of indexing. She may assume some extracted concepts to be irrelevant, or unimportant, or might want to add some other concepts.


QuizPack - TALER - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://nala.exp.sis.pitt.edu:8080/quizpack/editQuestion.jsp?request=null&concepts=EXTRACT
Back Search Favorites Media
QuizGuide Authoring Tool
::Home ::Authoring ::System Management
::My Account : Logout
Extract
Edit
QuizPack - TALER - Microsoft Internet Explorer
PATTERNS
PROGRAMMING TECHNIQUES
STYLE
C programming
SYNTAX
EXPRESSIONS
STATEMENTS
VARIABLES
PROGRAM STRUCTURE
PLATFORMS
WINDOWS
UNIX
APPLE
→HyperGraph
<<Add
Del>>
Submit resutls

New Question

Title : do-1(2)

Description : simple do loop

Code :

```
main ()
{
  int i = 0;
  do
    i++;
  while (i < 7);
}
```

Concepts :
main-function
int
initialization
do
post-increment
less

Edit Concepts

main-function
int
initialization
do
compound
post-increment
less

if-else-if
assign-min
assign-max
sentinel-loop
flag-loop
power
search-array
sort-array
read-array
print-array

Figure 1. Concept Elicitation from the Code of a Sample Question.

In other words, our intention was to develop a system which supports the authoring of intelligent content according to a teacher's preferences while maximally facilitating this process, but not impose an outside vision of the domain. To ensure this degree of flexibility, our system provides the author with a supplementary interface for editing the extracted list of concepts, or s/he may even create this list from scratch. To start this process an author needs to click on the button *Edit* in the *Concepts* section of the interface. A window loads, where an author can add or remove concepts from the index, either by using the lists of elicited (left) and available (left) concepts or by browsing the domain ontology.

The developed ontology of C programming contains about 150 concepts. About 30 of them are meta-concepts; their titles are written in black font. An author cannot add meta-concepts to the index and may use them only for navigational purposes. Leaves of the ontology can be either in the index or in the list of available concepts. First, they are represented by blue font on the white background, second, they are written in the light-blue squares. By clicking on leaves of the ontology an author adds (or removes if had already been added) a corresponding concept to the index: the background of the node in the ontology is changed and the concept moves from one list to another. The set of ontology leaves is a superset for the number of concepts available for automatic extraction. Figure 1 demonstrates the process that happens when an author wants to add a concept to the generated list. The parsing component has identified the concept "main-function" in the code of sample example. The compound operator is syntactically a part of the function definition, though the parser has not identified it as a separate concept. However, a teacher might want to stress that this is a particular case of compound operator and add this component by hand. As you can see, the index lists on the main window and on the window of the manual concept elicitation are different. The concept "compound" is added to the index manually, but is not saved at the moment. Hence, an author has freedom: s/he can choose to rely on the automatic indexing or can perform more precise manual indexing that best fits her/his needs.

As an ontology visualization tool we use the hypergraph software (http://hypergraph.sourceforge.net/), which provides an open source easy-tuneable platform for manipulating hyperbolic trees [15]. A number of research and practical projects are conducted currently on different types of tools for the visualization of large concept structures [16; 17]. Hyperbolic trees allow one to shift the focus away from unnecessary information while preserving the entire structure of the tree (or its sufficient part) on the screen. Since, our choice of ontology type is a simple taxonomy, tree structure is the best choice for representing the relationships of the domain concepts and organizing them into helpful navigational components.

3. Prerequisite/Outcome Identification

The outcomes of the concept elicitation stage are concept lists for all content elements (in this case, questions). However, prerequisite-based adaptive navigation support technique that we apply [7] requires all concepts associated with a content element to be divided into prerequisite and outcome concepts. Prerequisites are the concepts that students need to master before starting to work with the element. Outcomes denote concepts that are being learned in the process of work with the element.

We use an original algorithm for the automatic identification of prerequisite and outcome concepts for each element. This algorithm is also collaborative because it takes into account a specific way of teaching the course provided by the instructor. The source of knowledge for this algorithm is a sequence of learning goals defined by the instructor [18]. Each goal typically represents a course lecture. To define a new goal an instructor simply needs to group together all content elements that support a specific lecture. The result of this process is a sequence of groups of content elements that corresponds to a course-specific sequence of lectures. The prerequisite/outcome separation algorithm starts with the first lecture and works iteratively through the sequence of lectures.

- All concepts associated with content elements that form the first group (first lecture) are declared the outcomes of the first lecture and are marked as outcomes in the index of all content elements that form the first group.
- All concepts associated with content elements that form the second group (second lecture) are divided into lecture outcomes and lecture prerequisites. All concepts already listed as outcomes of the first lecture are defined as prerequisites of the second lecture. They are marked as prerequisite concepts for each content element in the second group. The concepts that were first mentioned in the second group become

outcomes of the second lecture. They are marked as outcome concepts for each content element in the second group.

- This process is repeated for each following group. On each step we separate concepts that are newly introduced and concepts that were introduced in one of the earlier lectures. The result of the process is a separation of prerequisite and outcome concepts for each lecture and each listed content element. A by-product of this process is the identification of the learning goal (a set of introduced concepts) of each lecture. Note that for each concept there is exactly one "home lecture" that introduced this concept.

Once the content elements are indexed and the goal sequence is constructed, any future additional element can be properly indexed and associated with a specific lecture in the course. The element is to be associated with the last lecture that introduces its concepts (i.e., the latest lecture, whose learning goal contains least one concept belonging to this element's index). After that, the element is associated with this lecture. It is important to stress again that the outcome identification is adapted to a specific way of teaching a course, as it is *mined* from the original sequence of content elements. It is known that different instructors teaching the same programming course may use a very different order for their concept presentation. Naturally, content sequencing in a course should be adapted to the instructor's preferred method of teaching. This is in contrast to the case when a teacher willing to use an adaptive system with the side-authored content in the class is forced to adjust the course structure to the system's view on it, or more precisely, to the view of the authors of the system.

4. Discussion and Future Work

This paper focuses on a new generation of authoring tools that support teachers as authors on intelligent content. We have presented a specific authoring system for automated collaborative indexing of parameterized questions. Although, some part of the system (the described automated approach to concept extraction, using a parsing component), is specific for the learning content based on the programming code (questions and code examples), we believe that the proposed general idea is applicable for a broad class of domains and content types. In less formalized domains, where concepts do not have a salient grammatical structure, the classic information retrieval approach could be used instead of parsing. The other two key ideas: ontology-based authoring support and prerequisite-outcome identification are domain independent.

The presented approach to intelligent content authoring as well as the implemented interface need exhaustive evaluation. Several research questions may arise:

- Does the proposed algorithm for prerequisite/outcome identification and concept elicitation provide good source for adequate adaptation?
- How helpful will the approach and the tool be for an arbitrary teacher, in indexing her/his own content?
- Are authors going to use the manual concept elicitation or will they stick to the automatic indexing? In the former case, will they prefer ontology-based authoring or simply turn to list manipulation?
- Are teachers going to take the time to author the adaptive content?

At the moment of writing we have formally evaluated one interactive component of the system – the concept-indexing tool based on hyperbolic trees. This component was evaluated in the context of a different authoring tool - Collaborative Paper Exchange [19]. The users of this tool are required to write summaries of research papers and index each summary with domain concepts. A short study presented in [19] evaluated the usability of the tool and compared two approaches to ontology-based indexing – traditional approach based on list selection and hyperbolic tree indexing. While the study showed that the current version of hyperbolic tree indexing is far from perfection, nine out of 14 subjects preferred hyperbolic tree indexing over traditional list-based indexing.

We will continue the evaluation process using several interactive tools we have developed for different types of learning activities. Our ultimate goal is to involve teachers into practical use of these tools and perform both subjective analysis of usability and objective evaluation of the labor-intensiveness of adaptive instruction authoring.

References

[1] Brusilovsky, P. and Peylo, C. Adaptive and intelligent Web-based educational systems. International Journal of Artificial Intelligence in Education, 13, 2-4 (2003), 159-172.

[2] Brusilovsky, P. and Miller, P. Course Delivery Systems for the Virtual University. In: Tschang, T. and Della Senta, T. (eds.): Access to Knowledge: New Information Technologies and the Emergence of the Virtual University. Elsevier Science, Amsterdam, 2001, 167-206.

[3] Ritter, S., Anderson, J., Cytrynowicz, M., and Medvedeva, O. Authoring Content in the PAT Algebra Tutor. Journal of Interactive Media in Education, 98, 9 (1998), available online at http://www-jime.open.ac.uk/98/9/.

[4] Koedinger, K.R., Anderson, J.R., Hadley, W.H., and Mark, M.A. Intelligent tutoring goes to school in the big city. In: Greer, J. (ed.) Proc. of AI-ED'95, 7th World Conference on Artificial Intelligence in Education, (Washington, DC, 16-19 August 1995), AACE, 421-428.

[5] Arroyo, I., Schapira, A., and Woolf, B.P. Authoring and sharing word problems with AWE. In: Moore, J.D., Redfield, C.L. and Johnson, W.L. (eds.) Artificial Intelligence in Education: AI-ED in the Wired and Wireless Future. IOS Press, Amsterdam, 2001, 527-529.

[6] Murray, T. Authoring Intelligent Tutoring Systems: An analysis of the state of the art. International Journal of Artificial Intelligence in Education, 10 (1999), 98-129, available online at http://cbl.leeds.ac.uk/ijaied/abstracts/Vol_10/murray.html.

[7] Brusilovsky, P., Eklund, J., and Schwarz, E. Web-based education for all: A tool for developing adaptive courseware. Computer Networks and ISDN Systems. 30, 1-7 (1998), 291-300.

[8] De Bra, P. and Calvi, L. AHA! An open Adaptive Hypermedia Architecture. The New Review of Hypermedia and Multimedia, 4 (1998), 115-139.

[9] Henze, N. and Nejdl, W. Adaptation in open corpus hypermedia. International Journal of Artificial Intelligence in Education, 12, 4 (2001), 325-350, available online at http://cbl.leeds.ac.uk/ijaied/abstracts/Vol_12/henze.html.

[10] Carmona, C., Bueno, D., Guzman, E., and Conejo, R. SIGUE: Making Web Courses Adaptive. In: De Bra, P., Brusilovsky, P. and Conejo, R. (eds.) Proc. of Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002) Proceedings, (Málaga, Spain, May 29-31, 2002), 376-379.

[11] Sosnovsky, S., Shcherbinina, O., and Brusilovsky, P. Web-based parameterized questions as a tool for learning. In: Rossett, A. (ed.) Proc. of World Conference on E-Learning, E-Learn 2003, (Phoenix, AZ, USA, November 7-11, 2003), AACE, 309-316.

[12] Lutz, R. Plan diagrams as the basis for understanding and debugging pascal programs. In: Eisenstadt, M., Keane, M.T. and Rajan, T. (eds.): Novice programming environments. Explorations in Human-Computer Interaction and Artificial Intelligence. Lawrence Erlbaum Associates, Hove, 1992, 243-285.

[13] Barr, A., Beard, M., and Atkinson, R.C. The computer as tutorial laboratory: the Stanford BIP project. International Journal on the Man-Machine Studies, 8, 5 (1976), 567-596.

[14] Weber, G. Individual selection of examples in an intelligent learning environment. Journal of Artificial Intelligence in Education, 7, 1 (1996), 3-31.

[15] Lamping, R. and Pirolli, P. A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. In: Katz, I., Mack, R. and Marks, L. (eds.) Proc. of CHI'95, (Denver, May 7-11, 1995), ACM, 401-408.

[16] Uther, J. and Kay, J. VlUM, a Web-based visualization of large user models. In: Brusilovsky, P., Corbett, A. and Rosis, F.d. (eds.) User Modeling 2003. Lecture Notes in Artificial Intelligence, Vol. 2702. Springer Verlag, Berlin, 2003, 198-202.

[17] Ziegler, J., Kunz, C., Botsch, V., and Schneeberger, J. Visualizing and exploring large networked information spaces with Matrix Browser. In: Proc. of 6th International Conference on Information Visualisation, IV'02, (London, UK, July 10-12, 2002), IEEE Computer Society.

[18] Brusilovsky, P. Developing Adaptive Educational Hypermedia Systems: From Design Models to Authoring Tools. In: Murray, T., Blessing, S. and Ainsworth, S. (eds.): Authoring Tools for Advanced Technology Learning Environments: Toward cost-effective adaptive, interactive, and intelligent educational software. Dordrecht, Kluwer, 2003, 377-409.

[19] Yudelson, M. and Brusilovsky, P. Collaborative Paper Exchange. In: Proc. of World Conference on E-Learning, E-Learn 2005, (Vancouver, Canada, November, 2005), AACE, submitted.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Some Unusual Open Learner Models

Susan BULL, Abdallatif S. ABU-ISSA, Harpreet GHAG & Tim LLOYD

Electronic, Electrical and Computer Engineering,
University of Birmingham, Edgbaston, Birmingham B15 2TT, U.K.
s.bull@bham.ac.uk

Abstract. Open learner models to facilitate reflection are becoming more common in adaptive learning environments. There are a variety of approaches to presenting the learner model to the student, and for the student to interact with their open learner model, as the requirements for an open learner model will vary depending on the aims of the system. In this paper we extend existing approaches yet further, presenting three environments that offer: (i) haptic feedback on learner model data; (ii) a handheld open learner model to support collaboration amongst mobile learners; (iii) an approach which allows students to open their model to selected or to all peers and instructors, in anonymous or named form.

1. Introduction

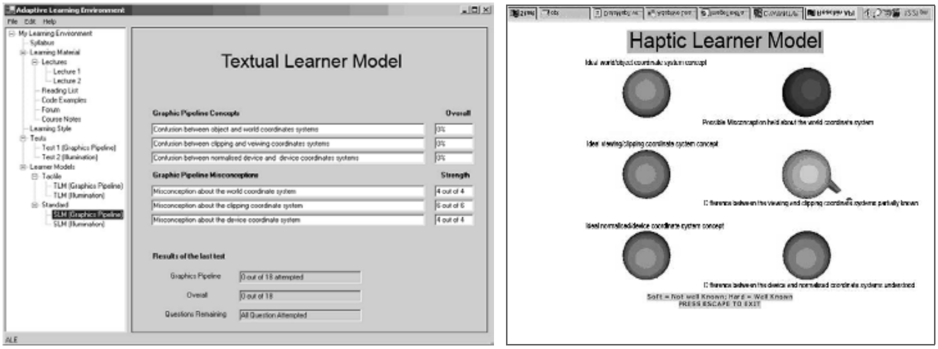
Open learner models - learner models that are accessible to users - are becoming more common in adaptive learning environments, to afford learners greater control over their learning [1] and/or promote reflection [2]. The simplest and most common is a skill meter, displaying a learner's knowledge as a subset of expert knowledge in part-filled bars showing progress in different areas [3]; or the probability that a student knows a concept [4]. Extensions to this include: skill meters showing a user's knowledge level compared to the combined knowledge of other user groups [5]; knowledge level as a subset of material covered which is, in turn, a subset of expert knowledge [6]; knowledge level as a subset of material covered, as a subset of expert knowledge, and also the extent of misconceptions and size of topic [7]. More detailed presentations allow specific concepts, and sometimes specific misconceptions held, to be presented to the learner; and/or relationships between concepts to be shown. This may be in a variety of formats, such as a hierarchical tree structure [1]; conceptual graph [8]; externalisation of connections in a Bayesian model [9]; textual description of beliefs [2]. This variety of methods of viewing learner models illustrates that there is no agreed standard or best approach to opening them to users. In addition to the varied methods of presenting models, there are different ways of interacting with them. For example, a learner may simply be able to view their model [4,6]; they may be able to edit (i.e. directly change) the contents [1,7]; or undertake a process of negotiation where student and system come to an agreement over the most appropriate representations for the learner's current understanding [2,8]. The choice of viewing and interaction methods depends on the system aims. Most open learner models are for access only by the student modelled. However, some systems also open the model to peers [10] or instructors [11].

In line with these varied approaches, we now extend the range yet further. We present three open learner models that go beyond the approaches of existing examples, by offering unique methods of using or interacting with the model. The first provides haptic feedback on the learner model contents. The second is for use on a handheld computer, with a simple model that can be carried around routinely, to facilitate peer collaboration should students come together opportunistically or for planned study sessions. The final example allows a learner to view the contents of their learner model, and also to open it to (selected or all) peers and (selected or all) instructors, either anonymously or with their names.

A survey of 44 university students found that students would be interested in using an open learner model. In particular, they want access to information about known topics or concepts (37 students), problems (40) and, perhaps most interesting because students often do not receive this information explicitly, identification of misconceptions (37) [12]. This was a survey-based investigation rather than an observation of system use, but similar results were later found amongst a group of 25 who had used an open open learner model that offers different views on the model data (extended version of [13]). 23 of the 25 found each of the above types of learner model information useful. In this paper we examine three quite different open learner modelling systems that model these attributes.

2. An Open Learner Model with Haptic Feedback

The haptic learner model is part of an environment that recommends material (slides, course notes, example code, exercises, discussion forum, further reading) on computer graphics according to the contents of the learner model constructed based on answers to multiple choice and item ordering questions. The learner model externalises to the user: concepts known, misconceptions as inferred from a misconceptions library, and difficulties inferred from incorrect responses that cannot be matched with specific misconceptions. Strength of evidence for knowledge and misconceptions is also given.



Textual Learner Model

Graphic Pipeline Concepts	Overall
Confusion between object and world coordinates systems	0%
Confusion between clipping and viewing coordinates systems	0%
Confusion between normalised device and device coordinates systems	0%

Graphic Pipeline Misconceptions	Strength
Misconception about the world coordinate system	4 out of 4
Misconception about the clipping coordinate system	6 out of 6
Misconception about the device coordinate system	4 out of 4

Results of the last test

Graphics Pipeline: 0 out of 18 attempted

Overall: 0 out of 18

Questions Remaining: All Question Attempted

Haptic Learner Model

Ideal world/object coordinate system concept

Possible Misconception held about the world coordinate system

Ideal viewing/clipping coordinate system concept

Difference between the viewing and clipping coordinate systems partially known

Ideal normalised/device coordinate system concept

Difference between the device and normalised coordinate systems understood

Soft = Not well Known; Hard = Well Known
PRESS ESCAPE TO EXIT

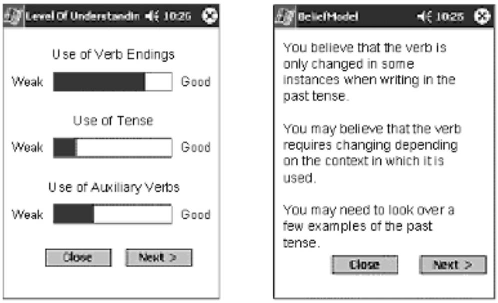
Fig. 1. A haptic learner model

There are two methods of accessing the model: a textual description (left of Fig. 1), and a version that combines text, graphics and haptic feedback (right of Fig. 1). Each allows access to the same information as described above. The textual model is straightforward, listing concepts and misconceptions, with a numerical indication of the strength of evidence for learner model entries. The haptic version displays a 3D scene with 'concept spheres' (with a textual description of the concept), which allow the learner to view and physically interact with their learner model using a haptic feedback device. The left side of the screen shows 'control spheres', indicating the state that learners are aiming for at their present stage of learning. The spheres to the right represent the learner's degree of understanding of the concepts on the left. Concepts are presented in shades of green - the brighter, the greater the level of understanding; and orange where the learner has difficulties. Misconceptions are red. As stated above, learners interact with their learner model using a haptic feedback device which provides force feedback. The haptic properties of the spheres are hard for concepts that are known well, and softer for less well-known concepts. Misconceptions also use the property of magnetism (or stickiness) in order to highlight the problem by physically drawing the user towards the sphere, leaving misconceptions feeling 'soft and sticky'.

20 3rd/4th year undergraduates studying computer engineering or computer science took part in a lab-based study to discover whether students are able to understand a haptic learner model, and whether they find it useful. Post-interaction questionnaires/interviews revealed that, of the 20, 12 found the haptic model intuitive, understanding its purpose; and the same number found it a useful support for their learning, with 11 finding it a useful means of encouraging reflection. 10 students found the textual and haptic versions equally useful, but 8, a large minority, found the haptic model more helpful. Students were also asked to self-diagnose their preferred approaches to learning before using the system. Of these, 10 claimed physical interaction and touch were important (as opposed to hearing, reading, watching). However, only 4 of these 10 were amongst those who preferred the haptic version of the learner model. Thus it appears that additional haptic feedback on learner model data could be useful, including for some who would not expect physical interaction to be helpful. This accords with findings in the context of viewing the learner model, that students have differing preferred presentations that are not related to learning style [13].

3. An Open Learner Model to Support Collaboration on the Move

Our second example is part of an environment for use on a handheld computer when students have short periods of time that they could not otherwise use for individualised interactions, such as on public transport, waiting for friends at a restaurant, etc. A model of the learner's knowledge, difficulties and misconceptions is created during an interaction in which students answer multiple choice English grammar questions following tutoring. The learner model is open for learner viewing as a standard part of the interaction, to help learners become more aware of their progress. In contrast to the previous system, our mobile open learner model is quite simple, as displayed in Fig. 2. It uses standard skill meters to indicate overall understanding of topics, with additional textual descriptions. The aim is *not* to present learners with all the details of their problems, but rather, to encourage them to think about their knowledge and difficulties, and develop or improve the metacognitive skills of self-monitoring and self-evaluation. Thus, the textual information provided, focuses on their beliefs and not the correctness (or otherwise) of those beliefs. It is the responsibility of the student to compare their learner model to the domain content.



You believe that the verb is only changed in some instances when writing in the past tense.

You may believe that the verb requires changing depending on the context in which it is used.

You may need to look over a few examples of the past tense.

Fig. 2. A mobile learner model to support collaboration

It is intended that learners not only reflect on their learner model individually, but a major purpose of the system is that students should routinely carry their learner models with them on their handheld computers, in order that they may compare them to the models of their friends if they meet opportunistically or for planned study sessions. Previous work suggested that students may engage in spontaneous peer tutoring if collaboratively examining their respective learner models [10]. This mobile version is intended to facilitate this process, as students do not have to meet in a fixed location where equipment is available, and do not necessarily have to schedule a learning session in advance.

The mobile learner model is part of an environment to teach English as a foreign language to advanced learners (e.g. university students in an English speaking country), who have difficulties with some aspects of grammar. Participants in the study described below were 8 Chinese MSc students at the University of Birmingham and 3 Punjabi-speaking students visiting Birmingham. The aim was simply to observe the way in which the system would be used in a semi-authentic setting. (The authenticity was necessarily limited by the presence of the experimenter and the need for video recordings for evaluation purposes.) There were no differences observed between the groups. The Chinese students arranged to meet for a meal at the home of one of the students, to combine a social occasion with a study session. The evaluation with the Punjabi students took place where one of them was staying, during a planned study session. Students joined together in pairs (in the case of the Punjabi students, a group of 3), and compared their learner models. They were given no instruction on how to approach discussion, or what to talk about. The following excerpt from one of the paired dialogues illustrates the kind of discussions that took place (transcribed from video recordings), when viewing the textual model descriptions:

S5: "Do you know what the past perfect continuous is? I am very confused, I do not understand. Is it used to talk about something that happened...well, I am not sure."

S3: "I think it is used to describe something that has happened before you do something else, so when you talk about two things. What score did you get for it?"

This illustrates that students are able to identify their areas of difficulty from their learner model, and will explain the grammar rules to each other. The final comment indicates that students were using their respective levels of performance shown by the skill meters, to decide which of them is more likely to be using a rule correctly, and hence able to explain it to the other. Other comments from the paired interactions include the following, further illustrating the common focus on correctness as portrayed in the learner model skill meters:

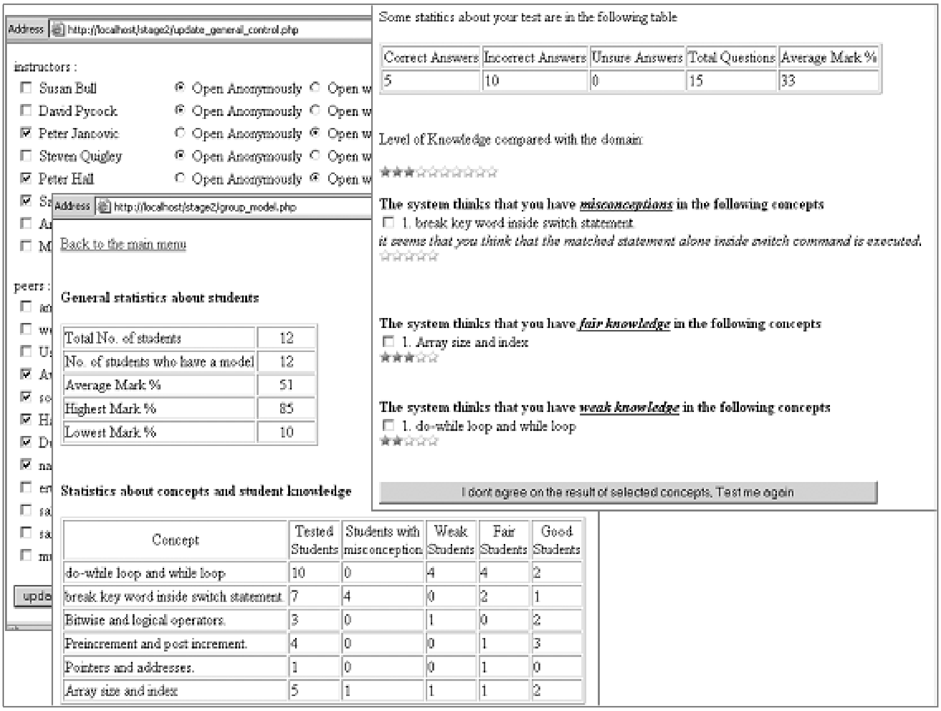
"I did not do so good in the past perfect. What did you get for that?"

"You do better in the past perfect, can you tell me what it is? I did not do well on that."

Students were willing to discuss their models. However, given that performance levels were available and seemed to be a focus of discussion, we would consider *not* providing such information (i.e. not using skill meters). Students would then have to think more about their beliefs to decide who may be best able to explain a rule in cases where their models differ (i.e. knowledge or specific problems rather than knowledge level). This would fit better with the aim of developing the skill of self-evaluation. It might result in a greater degree of reflection: in a context where information about level of performance was not given, students thought more carefully about their respective beliefs, and spontaneous peer tutoring was observed [10]. It would therefore be interesting to compare discussion and learning outcomes of students who have the skill meters and students who do not. A further issue to consider is how the absence of skill meters might affect individual use.

4. A Learner Model that can be Opened to Peers and Instructors

We now return to the desktop PC, with an open learner model showing knowledge level of C programming in skill meter form (as a series of filled and unfilled stars), and a corresponding textual description, constructed based on responses to multiple choice questions (Fig. 3). A statement of misconceptions inferred from a misconceptions library is also presented. If the learner disagrees with the model, they can request a test to quickly update it. Students can open their model to peers and/or instructors, choosing for each individual whether to release their model anonymously or with their personal details. Peer models are accessed by clicking on a peer's name or model number (for models released anonymously). Note that some learners may access a peer model anonymously, while others have named access to the same model, and yet others have no access. Students can view group data of the distribution of knowledge/problems across all users.


Address http://localhost/stage2/update_general_control.php

instructors :

□ Susan Bull ● Open Anonymously ○ Open w
□ David Pycock ● Open Anonymously ○ Open w
☑ Peter Jancovic ○ Open Anonymously ● Open w
□ Steven Quigley ● Open Anonymously ○ Open w
☑ Peter Hall ○ Open Anonymously ● Open w
☑ Sa
□ Ar
□ M

peers :

□ an
□ we
□ Us
☑ Ar
☑ so
☑ Ha
☑ De
☑ na
□ en
□ sa
□ sa
□ m

upda

Address http://localhost/stage2/group_model.php

Back to the main menu

General statistics about students

Total No. of students	12
No. of students who have a model	12
Average Mark %	51
Highest Mark %	85
Lowest Mark %	10

Statistics about concepts and student knowledge

Concept	Tested Students	Students with misconception	Weak Students	Fair Students	Good Students
do-while loop and while loop	10	0	4	4	2
break key word inside switch statement	7	4	0	2	1
Bitwise and logical operators.	3	0	1	0	2
Preincrement and post increment.	4	0	0	1	3
Pointers and addresses.	1	0	0	1	0
Array size and index	5	1	1	1	2

Some statitics about your test are in the following table

Correct Answers	Incorrect Answers	Unsure Answers	Total Questions	Average Mark %
5	10	0	15	33

Level of Knowledge compared with the domain

★★★☆☆☆☆☆☆☆

The system thinks that you have *misconceptions* in the following concepts

□ 1. break key word inside switch statement

it seems that you think that the matched statement alone inside switch command is executed.

☆☆☆☆☆

The system thinks that you have *fair knowledge* in the following concepts

□ 1. Array size and index

★★★☆☆

The system thinks that you have *weak knowledge* in the following concepts

□ 1. do-while loop and while loop

★★☆☆☆

I dont agree on the result of selected concepts. Test me again

Fig. 3. A learner model open to students, peers and instructors

Table 1. Opening the learner model to others

Student	*Open for Instructors*					*Open for Peers*				
	None	All		Selected		None	All		Selected	
		anon	named	anon	named		anon	named	anon	named
S1		X						X		
S2					X					X
S3			X					X		
S4		X					X			
S5	X						X			
S6			X				X			
S7				X		X				
S8		X						X		
S9					X		X			
S10			X					X		
S11		X					X			
S12			X					X		
Total	1	4	4	1	2	1	5	5	0	1

12 MSc students in Electronic, Electrical and Computer Engineering took part in an initial lab study to investigate whether students would be willing to open their models to

others and, if so, whether they would do so named or anonymously. Results are in Table 1. Only 1 student chose not to open their model to instructors, and 1 to peers. These were different students in each case. 8 opened their learner model to all instructors, 4 of whom did so anonymously, and 4 named. 3 opened their model to selected instructors only - 1 anonymously and 2 named. 1 student opened their learner model only to selected peers. 10 students opened their model to all peers, 5 anonymously and 5 named. Those who opened their model anonymously to instructors did not necessarily choose to remain anonymous to peers, and those who allowed instructors to view their learner model with personal details did not necessarily allow peers to view their identifying data. This small-scale study has not allowed us to investigate possible patterns of opening the model over time - the aim at this initial stage was to determine whether students are willing to make their learner model data available to others, and whether they wish to view the models of peers. Usage suggests that providing a choice of how and to whom to open the learner model, is important. In a post-interaction questionnaire, 10 of the 12 students stated that being able to select between individuals was useful, and all 12 liked the anonymous/named distinction. 11 stated that they found their own learner model useful. 8 found the individual peer models useful, and 8 found the group model useful. Thus viewing their own learner model seemed to be useful for the majority, and peer models also appear helpful for many. Comparing questionnaire results to the usage data, the facility to make the choice of who should have access to their learner model seems important even for students who opened their model to everyone.

5. Discussion

The haptic learner model was designed for individual users who prefer physical interaction in learning to encourage their interest in the learner model, but it may also be perceived as useful by others. However, longer term use needs to be studied to determine the extent to which positive reactions are related to the novelty of the haptic approach. The other two systems are essentially individual environments with learner models that can also be viewed by other people. Learners who enjoy collaboration and the social side of learning may favour the mobile environment, which expects co-present peers. However, the collaborative phase is not essential, and the system could be used simply in situations where the learner is away from a desktop PC. The final example was designed specifically for a broader range of students - those who like to work individually, who may or may not wish to compare their learner model with models of peers; those who enjoy collaborative learning who may use the peer models to seek learning partners; or competitive learners who strive to outperform others, who may check their progress against peers, without interacting with those other students. While the above descriptions of learner types match some of the learner groups described by various learning style categorisations (of which there are many), we do not wish to prescribe certain interaction methods for different learners according to their learning style, until more is understood about the relationship between learning style and computer-based educational interactions, including methods of access to open learner models, as a clear relationship between the two cannot be assumed [see 13].

While the underlying representations in our three systems are quite similar, the information available to learners differs. The haptic model only names the concepts and misconceptions, with an indication of the strength of each (by visual or haptic properties), but does not give further detail. The mobile open learner model presents an overview of the extent of understanding, together with a textual description of beliefs, but without ascribing any level of correctness to the textual information. Thus students know their general level of ability or skill, but must themselves determine the specific details of what they know, or what their problems may be. The model that can be opened to peers and instructors lists concepts known and specific misconceptions, and allows group data to be displayed, which can be compared to individual performance. Each of the open learner models was designed to fit the purpose for which it was created, which necessarily results in these differences.

While some previous findings suggest students may not use open learner models [14,15], results are more positive for studies where the open learner model was integrated into the interaction [2,8]. Initial evaluations of the systems in this paper have indicated that more unusual approaches to integrated open learner models may also be of benefit. However, it is not expected that each of the approaches will suit all learners. Adaptive learning environments came into being because of the recognition that learners are different, and the function of these systems is to adapt to individual differences. There is no reason to suppose that use of an open learner model is any different - students may differentially benefit from the existence of an open learner model, and also from the method of viewing, sharing and interacting with it. Our aim, then, is to further develop open learner models that are useful to sufficient numbers of learners to make this worthwhile. It is likely that this will often involve models that can be viewed or accessed in different ways, rather than the more common single learner model presentation in most current systems. It has been found that students have clear preferences for how to view their learner model [13]. The three systems in this paper illustrate this to some extent. The mobile learner model can be viewed as a skill meter overview or as a more detailed textual description of beliefs, though it is likely that learners will use both. (However, as noted above, we would consider removing the skill meters, as one of the aims of the environment is to develop the metacognitive skill of self-evaluation. The skill meters may stifle this in a collaborative setting.) Regardless of whether the skill meters are maintained, the main difference in usage will probably be in whether students use the model individually, or as part of a collaborative session. This is also true of the system that allows learners to open their model to others. With our small group, most students opened their learner model to all peers. In a recent study with 50 students, initial findings are that some learners open their models quite widely, while some prefer a more restricted focus amongst those they know well, or even an individual focus. Most students viewed the peer models positively, using them to find their relative position in the class and which topics are generally difficult. Some used them to seek collaborators, while some used them competitively, to try to outperform others [16]. The haptic model may be accessed differentially, either the textual or haptic version, since these show the same information.

The evaluations described in this paper are, of course, quite limited, and should be regarded only as a first step. Further work is required to answer questions such as:

- When the haptic learner model is no longer a novelty, will students continue to use it?
- Will a haptic learner model work best in a learning environment that uses haptic interaction in other areas, or can it be equally useful in an environment that otherwise uses no force-feedback?
- Will students really use their mobile learner models when they meet opportunistically, or might they be used only when collaborative learning sessions have been planned?
- Would removing the mobile skill meters result in more reflective discussion?
- Would removal of the skill meters be beneficial or detrimental to individual usage?
- To what extent will learners use the models of peers over an extended period?
- Will instructors really use the information about their students, or would other demands on their time make this unlikely in practice?
- Is there any difference in performance with different kinds of open learner model, or does the effect of the presentation or interaction method vary according to the individual's preferences? To what extent is this presentation or preference-specific?

There remain many issues to address before we may discover the real potential of such unusual open learner models, but initial results suggest that this research is worth pursuing.

6. Summary

There are many approaches to opening the learner model to the learner, and there is no agreed or best method for doing so. Requirements for open learner models are dependent on the aims of the systems in which the models are used. This paper has broadened the approaches to open learner modelling yet further, with three new examples. Early work has suggested that further investigation of extensions to existing open learner modelling approaches is worthwhile, and it has been suggested that systems might benefit from allowing users to view and/or interact with their learner model in different ways.

References

[1] Kay, J. (1997). Learner Know Thyself: Student Models to Give Learner Control and Responsibility, *Proceedings of International Conference on Computers in Education*, AACE, 17-24.

[2] Bull, S. & Pain, H. (1995). 'Did I say what I think I said, and do you agree with me?': Inspecting and Questioning the Student Model, *Proceedings of World Conference on Artificial Intelligence in Education*, AACE, Charlottesville, VA, 1995, 501-508.

[3] Weber, G. & Brusilovsky, P. (2001). ELM-ART: An Adaptive Versatile System for Web-Based Instruction, *International Journal of Artificial Intelligence in Education* 12(4), 351-384.

[4] Corbett, A.T. & Bhatnagar, A. (1997). Student Modeling in the ACT Programming Tutor: Adjusting a Procedural Learning Model with Declarative Knowledge, *User Modeling: Proceedings of 6th International Conference*, Springer Wien New York, 243-254.

[5] Linton, F. & Schaefer, H-P. (2000). Recommender Systems for Learning: Building User and Expert Models through Long-Term Observation of Application Use, *User Modeling and User-Adapted Interaction* 10, 181-207.

[6] Mitrovic, A. & Martin, B. (2002). Evaluating the Effects of Open Student Models on Learning, *Adaptive Hypermedia and Adaptive Web-Based Systems, Proceedings of Second International Conference*, Springer-Verlag, Berlin Heidelberg, 296-305.

[7] Bull, S. & McEvoy, A.T. (2003). An Intelligent Learning Environment with an Open Learner Model for the Desktop PC and Pocket PC, in U. Hoppe, F. Verdejo & J. kay (eds), *Artificial Intelligence in Education*, IOS Press, Amsterdam, 389-391.

[8] Dimitrova, V. (2003). StyLE-OLM: Interactive Open Learner Modelling, *International Journal of Artificial Intelligence in Education* 13(1), 35-78.

[9] Zapata-Rivera, J-D. & Greer, J.E. (2004). Interacting with Inspectable Bayesian Student Models, *International Journal of Artificial Intelligence in Education* 14(2), 127-163.

[10] Bull, S. & Broady, E. (1997). Spontaneous Peer Tutoring from Sharing Student Models, in B. du Boulay & R. Mizoguchi (eds), *Artificial Intelligence in Education*, IOS Press, Amsterdam.

[11] Mühlenbrock, M., Tewissen, F. & Hoppe, H.U. (1998). A Framework System for Intelligent Support in Open Distributed Learning Environments, *International Journal of Artificial Intelligence in Education* 9(3-4), 256-274.

[12] Bull, S. (2004). Supporting Learning with Open Learner Models, *Proceedings of 4th Hellenic Conference in Information and Communication Technologies in Education*, Athens, 47-61.

[13] Mabbott, A. & Bull, S. (2004). Alternative Views on Knowledge: Presentation of Open Learner Models, *Intelligent Tutoring Systems: 7th Int. Conference*, Springer-Verlag, Berlin Heidelberg, 689-698.

[14] Barnard, Y.F. & Sandberg, J.A.C. (1996). Self-Explanations, do we get them from our students?, *Proceedings of European Conference on Artificial Intelligence in Education*, Lisbon, 115-121.

[15] Kay, J. (1995). The UM Toolkit for Cooperative User Modelling, *User Modeling and User Adapted Interaction* 4, 149-196.

[16] Bull, S., Mangat, M., Mabbott, A., Abu Issa, A.S. & Marsh, J. (Submitted). Reactions to Inspectable Learner Models: Seven Year Olds to University Students, Submitted for publication.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Advanced Capabilities for Evaluating Student Writing: Detecting Off-Topic Essays Without Topic-Specific Training

Jill BURSTEIN
Derrick HIGGINS
Educational Testing Service
Princeton, New Jersey, USA

Abstract. We have developed a method to identify when a student essay is off-topic, i.e. the essay does not respond to the test question topic. This task is motivated by a real-world problem: detecting when students using a commercial essay evaluation system, *Criterion*^{SM}, enter off-topic essays. Sometimes this is done in bad faith to trick the system; other times it is inadvertent, and the student has cut-and-pasted the wrong selection into the system. All previous methods that perform this task require 200-300 human scored essays for training purposes. However, there are situations in which no essays are available for training, such as when a user (teacher) wants to spontaneously write a new topic for her students. For these kinds of cases, we need a system that works reliably without training data. This paper describes an algorithm that detects when a student's essay is off-topic without requiring a set of topic-specific essays for training. The system also distinguishes between two different kinds of off-topic writing. The results of our experiment indicate that the performance of this new system is comparable to the previous system that does require topic-specific essays for training, and conflates different types of off-topic writing.

Introduction

Research problems in text document classification include sorting of e-mail ([17],[8]) internet-based search engines ([15],[13]), automatic cataloguing of news articles, ([1],[3]) and classifying information in medical reports ([12],[20],[7]). Our research problem also relates to text classification, but in an educational domain: automated essay evaluation. Much work has been done in this area with regard to automated essay scoring ([16], [4],[10],[14],[9]). Our problem is a bit different. Specifically, our task is to evaluate if a student has written an *off-topic* essay ([6]).

The context of this work is the development of an off-topic essay detection capability that will function within *Criterion*^{SM}, a web-based, commercial essay evaluation system for writing instruction ([5]). *Criterion* contains two complementary applications. The scoring application*, e-rater*$^{®}$, extracts linguistically-based features from an essay and uses a statistical model of how these features are related to overall writing quality to assign a ranking (score) to the essay, typically on a scoring scale of 1 (worst) to 6 (best). The second application, *Critique*, is comprised of a suite of programs that evaluates errors in grammar, usage, and mechanics, identifies an essay's discourse structure, and recognizes undesirable stylistic features. *Criterion* currently has additional functionality that provides such feedback about off-topic writing to students. For training purposes, however, the current method requires a significant number (200-300) of human-reader scored essays that are written to a particular test question (topic). This can be problematic in the following situation. *Criterion* allows users (teachers) to spontaneously write new topics for their students. In addition, *Criterion* content developers may also add new topics to the system periodically. In both cases, there is no chance to collect and manually score 200–300 essay responses. Another weakness of the current method is that it addresses different kinds of off-topic writing in the same way.

In this study, we have two central tasks: First, we want to develop a method for identifying off-topic essays that does not require a large set of topic-specific training data, and secondly, we also want to try to develop a method that captures two different kinds of off-topic writing: *unexpected topic essays* and *bad faith essays*. The differences between these two are described below.

In the remaining sections of this paper, we will define what we mean by an off-topic essay, discuss the current methods used for identifying off-topic essays, and introduce a new approach that uses content vector analysis, but does not require large sets of human-scored essay data for training. This new method can also distinguish between two kinds of off-topic essays.

1. What Do We Mean By Off-Topic?

Though there are a number of ways to form an off-topic essay, this paper will deal with only two types. In the first type, a student writes a well-formed, well-written essay on a topic that does not respond to the expected test question. We will refer to this as the *unexpected topic* essay. This can happen if a student inadvertently cuts-and-pastes the wrong essay that s/he has prepared off-line.

In another case, students enter a *bad faith* essay into the application, such as the following:

> "*You are stupid. You are stupid because you can't read. You are also stupid becuase you don't speak English and because you can't add.*
>
> *Your so stupid, you can't even add! Once, a teacher give you a very simple math problem; it was 1+1=?. Now keep in mind that this was in fourth grade, when you should have known the answer. You said it was 23! I laughed so hard I almost wet my pants! How much more stupid can you be?!*
>
> *So have I proved it? Don't you agree that your the stupidest person on earth? I mean, you can't read, speak English, or add. Let's face it, your a moron, no, an idiot, no, even worse, you're an imbosol."*

Both cases may also happen when users just want to try to fool the system. And, *Criterion* users are concerned if either type is not recognized as off-topic by the system. A third kind of off-topic essay is what we call the *banging on the keyboard* essay, e.g., "*alfjdla dfadjflk ddjdj8ujdn.*" This kind of essay is handled by an existing capability in *Criterion* that considers ill-formed syntactic structures in an essay.[1] In the two cases that we consider, the essay is generally well-formed in terms of its structure, but it is written without regard to the test question topic. Another kind of off-topic writing could be a piece of writing that contains any combination of *unexpected topic*, *bad-faith*, or *banging on the keyboard* type texts. In this paper, we deal only with the *unexpected topic* and *bad-faith* essays.

2. Methods of Off-Topic Essay Identification

2.1 Computing Z-scores, Using Topic-Specific Essays for Training

[1] This method was developed by Thomas Morton.

In our current method of off-topic essay detection, we compute two values derived from a content vector analysis program used in *e-rater* for determining vocabulary usage in an essay ([5],[1]).[2] Off-topic in this context means that a new, unseen essay appears different from other essays in a training corpus, based on word usage, or, an essay does not have a strong relationship to the essay question text. Distinctions are not necessarily made between *unexpected topic* or a *bad faith* essays.

For each essay, *z-scores* are calculated for two variables: a) relationship to words in a set of training essays written to a prompt (essay question), and b) relationship to words in the text of the prompt. The *z-score* value indicates a novel essay's relationship to the mean and standard deviation values of a particular variable based on a training corpus of human-scored essay data. The score range is usually 1 through 6, where 1 indicates a poorly written essay, and 6 indicates a well-written essay. To calculate a *z-score*, the mean value and the corresponding standard deviation (SD) for *maximum cosine* or *prompt cosine* are computed based on the human-scored training essays for a particular test question.[3] For our task, z-*scores* are computed for: a) the *maximum cosine,* which is the highest cosine value among all cosines between an unseen essay and all human-scored training essays, and b) the *prompt cosine* which is the cosine value between an essay and the text of the prompt (test question). When a *z-score* exceeds a set threshold, it suggests that the essay is anomalous, since the threshold typically indicates a value representing an acceptable distance from the mean.

We evaluate the accuracy of these approaches based on the false positive and false negative rates. The *false positive rate* is the percentage of appropriately written, on-topic essays that have been incorrectly identified as off-topic; the *false negative rate* is the percentage of true off-topic essays not identified (missed) as off-topic. Within a deployed system, it is preferable to have a lower false positive rate. That is, we are more concerned about telling a student, incorrectly, that s/he has written an off-topic essay, than we are about missing an off-topic essay.

For the *unexpected topic* essay set[4], the rate of false positives using this method is approximately 5%, and the rate of false negatives is 37%, when the z-scores of both the *maximum cosine* and *prompt cosine* measures exceed the thresholds. For *bad faith* essays, the average rate of false negatives is approximately 26%.[5] A newer prompt-specific method has been developed recently that yields better performance. For proprietary reasons, we are unable to present the methods in this paper. For this proprietary method, the rate of false positives is 5%, and the rate of false negatives is 24%. For the *bad faith* essay data, the false negative rate was 1%. Unfortunately, this new and improved method still requires the topic-specific sets of human-scored essays for training.

2.2 Identifying Off-Topic Essays Using CVA & No Topic-Specific Training Data

An alternative model for off-topic essay detection uses content vector analysis (CVA)[6], and also relies on similarity scores computed between new essays and the text of the prompt on

[2] This method was developed and implemented by Martin Chodorow and Chi Lu.

[3] The formula for calculating the *z-score* for an new novel essay is: *z-score* = (*value* –*mean*) ÷ SD

[4] See *Data Section 2.2.2* for descriptions of the data sets.

[5] We cannot compute a false positive rate for the *bad faith* essays, since they are not written to any of the 36 topics.

[6] During the course of this study, we have experimented with applying another vector-based similarity measure to this problem, namely Random Indexing (RI) ([18]). Our results indicated that CVA had better performance. We speculate that the tendency of Random Indexing (RI), LSA, and other reduced-dimensionality vector-based approaches to assign higher similarity scores to texts that contain similar (but not

which the essay is supposed to have been written. Unlike the method described in Section 2.1, this method does not rely on a pre-specified similarity score cutoff to determine whether an essay is on or off topic. Because this method is not dependent on a similarity cutoff, it also does not require any prompt-specific essay data for training in order to set the value of an on-topic/off-topic parameter.

Instead of using a similarity cutoff, our newer method uses a set of *reference essay prompts*, to which a new essay is compared. The similarity scores from all of the essay-prompt comparisons, including the similarity score that is generated by comparing the essay to the target prompt, are calculated and sorted. If the target prompt is ranked amongst the top few vis-à-vis its similarity score, then the essay is considered on topic. Otherwise, it is identified as off topic.

This new method utilizes information that is available within Criterion, and does not require any additional data collection of student essays or test questions.

2.2.1 Content Vector Analysis

The similarity scores needed for this method of off-topic essay detection are calculated by content vector analysis. CVA is a vector-based semantic similarity measure, in which a content vector is constructed for the two texts to be compared, and their similarity is calculated as the cosine of the angle between these content vectors ([19]). Basically, texts are gauged to be similar to the extent that they contain the same words in the same proportion.

We do not do any stemming to preprocess the texts for CVA, but we do use a stoplist to exclude non content-bearing words from the calculation. We use a variant of the *tf*idf* weighting scheme to associate weights with each word in a text's content vector. Specifically, the weight is given as (1+log(tf))×log(D/df), where *tf* is the "term frequency", *df* is the "document frequency", and *D* is the total number of documents in the collection. The term frequencies in this scheme are taken from the counts of each word in the document itself, of course (the essay or prompt text). The document frequencies in our model are taken from an external source, however. Ideally, we could calculate how many documents each term appears in from a large corpus of student essays. Unfortunately, we do not have a sufficiently large corpus available to us, so instead, we use document frequencies derived from the TIPSTER collection ([11]), making the assumption that these document frequency statistics will be relatively stable across genres.

2.2.2 Data

Two sets of data are used for this experiment: *unexpected topic* essays and *bad faith* essays. The data that we used to evaluate the detection of *unexpected topic essays* contain a total of 8,000 student essays. Within these 8,000 are essays written to 36 different test questions (i.e., prompts or topics), approximately 225 essays per topic. The level of essay spans from the 6th through 12th grade. There is an average of 5 topics per grade. These data are all good faith essays that were written to the expected topic.[7] The data used to evaluate the detection of *bad faith essays* were a set of 732 essays for which a human reader has assigned a score of '0'. These 732 essays were extracted from a larger pool of approximately 11,000 essays that had received a score of '0.' Essays can receive a score of '0' for a number of reasons, including: the essay is blank, the student only types his or her

the same) vocabulary may be a contributing factor. The fact that an essay contains the exact words used in the prompt is an important clue that it is on topic, and this may be obscured using an approach like RI.

[7] Note, however, that on-topic essays for one prompt can be used as exemplars of unexpected-topic essays for another prompt in evaluating our systems.

name into the essay, the student has only cut-and-pasted the essay question, or the essay is off-topic. Of the 11,000, we determined that this set of 732 were *bad faith, off-topic* essays, using an automatic procedure that identified an extremely low percentage of words in common between the test question and the essay response. These essays were taken from a different population than the 6th through 12th grade essays. These were from a graduate school population. In addition, none of the essay questions these essays were supposed to respond to were the same as the 36 test questions in the 6th to 12th grade pool of essay questions. We also manually read through this set of 732 essays to ensure that they were *bad faith* essays as opposed to the *unexpected topic* type.

2.3 Evaluation & Results

2.3.1 Unexpected Topic Essays

We know from previous experimentation that essays tend to have a significant amount of vocabulary overlap, even across topics, as do the test questions themselves. For instance, if one topic is about '*school'* and another topic is about '*teachers*,' essays written to these topics are likely to use similar vocabulary. Even more generally, there is a sublanguage of essays that may be referred to as generic word use. In the sublanguage of standardized test essays are words, such as "I," "agree," and "opinion." Therefore, selecting a discrete threshold based on any measure to estimate similar vocabulary usage between an essay and the essay question has proven to be ineffective. Specifically, the similarity of essays to their (correct) prompt can be highly variable, which makes it impossible to set an absolute similarity cutoff to determine if an essay is on an unexpected topic. However, we can be fairly certain that the target prompt should at least rank among the *most* similar, if the essay is indeed on topic. Given this, we carried out the evaluation in the following way.

Starting with our 36 prompts (topics), we performed an 18-fold cross-validation. For each fold, we use 34 *reference prompts*, and two *test prompts*. This cross-validation setup allows us to distinguish two different evaluation conditions. The first, *training set performance,* is the system's accuracy in classifying essays that were written on one of the reference prompts. The second, *test set performance,* is the accuracy of the system in classifying essays which were written on one of the test prompts.

For each cross-validation fold, each essay from across the 34 reference prompts is compared to the 34 reference prompt texts, using the cosine correlation value from CVA. Therefore, an essay is compared to the actual prompt to which it was written, and an additional 33 prompts on a different, unexpected topic. Based on the computed essay-prompt cosine correlation value, essays are considered 'on-topic' only if the value is among the top *N* values; otherwise the essay is considered to be off-topic. So, for instance, if the similarity value is amongst the top 5 of 34 values (top 15%), then the essay is considered to be on-topic. This gives rise to the training set performance shown in Figure 1. The essays written to the test prompts are also evaluated. If A and B are the two test prompts, then all essays on prompt A are compared to the 34 reference essays and to prompt A, while all essays on prompt B are compared to the 34 reference essays and to prompt B. The resulting rankings of the prompts by similarity are used to determine whether each test essay is correctly identified as on-topic, producing the false positive rates for the training set in Figure 1. Finally, all essays on prompt A are compared to the 34 reference essays and to prompt B, while all essays on prompt B are compared to the 34 reference essays and to prompt A. This allows us to generate the false negative rates for the training set in Figure 1.

Figure 1 shows the tradeoff between the false positive rate and the false negative rate in our model of unexpected-topic essay detection. **The number labeling each point on the**

graph indicates the cutoff *N*, which is the number of prompts considered close enough to the essay to be regarded as on-topic. The best choice of this parameter for our application is probably around 10, which gives us a false positive rate of 6.8% and a false negative rate of 22.9% on test data. These rates represent only a moderate degradation in performance compared to the supervised methods described in *Section 3.1*, but are achieved without the use of labeled training data.

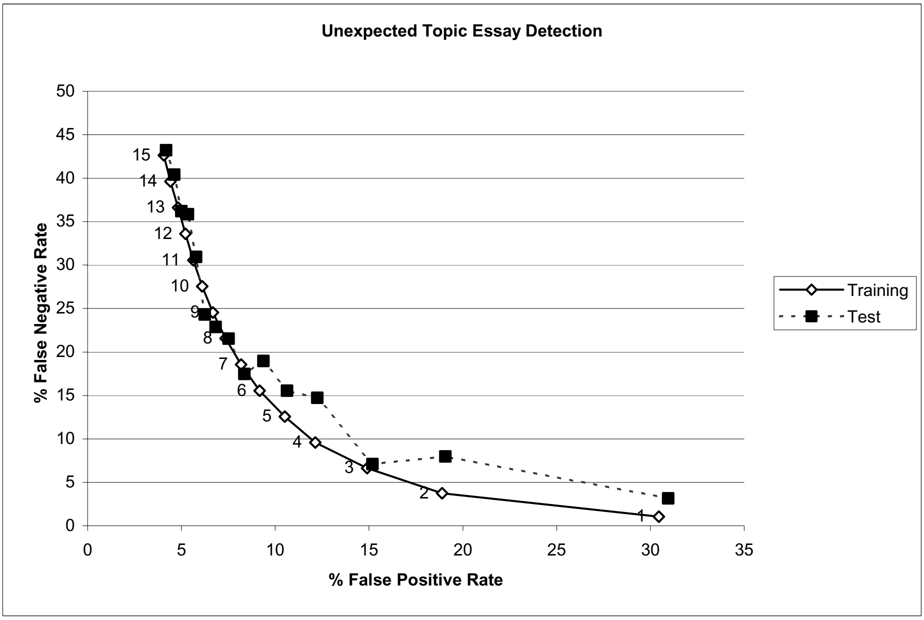


Figure 1: Performance of CVA-based model in predicting unexpected-topic essays

2.3.2 Bad Faith Essays

For identifying *bad faith* essays, it is more appropriate to use a similarity cutoff because we do not expect these essays to share much vocabulary with any prompt. These are the worst-case off-topic essays, where no attempt was made to answer any kind of essay question.

To evaluate this simple model for detecting bad-faith essays, we generated similarity scores each of the 36 prompts and each of the 732 known bad-faith essays. All essays whose CVA similarity scores with a prompt fell below a cutoff value were correctly identified as bad-faith. If we then count the essays from this set that were not identified as bad-faith, this gives us the false negative rates in Figure 2. Using the same cutoff values, we evaluated how many of the on-topic essays for each of the 36 prompts would be identified as bad-faith by this method. This resulted in the false positive rates in Figure 2. Performance outcomes for the *unexpected topic* and the *bad faith* essay detection evaluations are reported in Figure 2, for a range of similarity cutoff values. **Similarity cutoff values label selected points on the continuous graph that shows the tradeoff between false positives and false negatives. The best cutoff value for our application is probably around .005, which gives us a false positive rate of 3.7% and a false negative rate of 9.2%.**

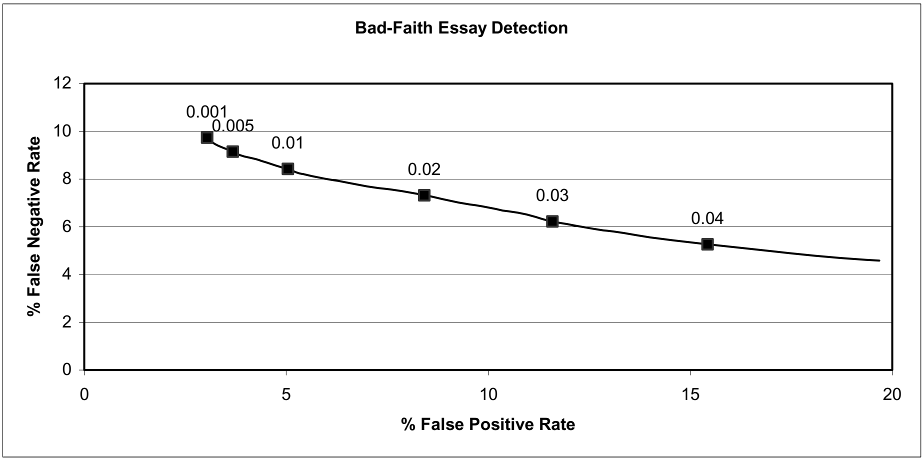


Figure 2: Performance of CVA-based model in predicting bad-faith essays

Discussion and Conclusions

Criterion^{SM} is an on-line essay evaluation service that has over 500,000 subscribers. Currently, the system has only a supervised algorithm for detecting off-topic essays input by student writers. Since this supervised method requires 200 – 300 human-scored essays to train each new essay question, the application can not provide feedback about off-topic writing for topics entered on-the-fly by instructors, and by the same token, if *Criterion* content developers want to periodically add new essay questions, off-topic essay detection cannot be applied until sufficient human-scored data are collected. In addition, the current supervised method treats all off-topic essays alike.

In this study, we have developed an unsupervised algorithm that requires only text of existing essay questions, the text of the new essay question, and the student essay in order to predict off-topicness. Our method also makes a distinction between two kinds of off-topic essays: *unexpected topic* and *bad-faith essays*. This new method uses content vector analysis to compare a new essay with the text of the essay to which it is supposed to be responding (target prompt), as well as a set of additional essay question texts. Based on these comparisons two procedures are applied. One procedure evaluates if the essay is on topic using the value between a new essay and the target prompt. If this value is amongst the highest CVA values, as compared to the values computed between the same essay and all other prompts, then the essay is on topic. If the essay-prompt comparison shows that the CVA value is not amongst the highest, then this method indicates with similar accuracy to the supervised method, that the essay is off topic, and also an *unexpected topic* essay. In the second procedure, a CVA value is selected that represents a lower threshold, based on a set of CVA essay-prompt comparisons. This lower threshold value represents an essay-prompt comparison in which the two documents contain little word overlap. If the CVA value computed between a new essay and the target prompt is equal to or lower than the pre-set lower threshold, then this is indicative of a *bad-faith* essay. In future work, we plan to look at additional kinds of off-topic writing.

Acknowledgements

We would like to thank Martin Chodorow, Paul Deane, Daniel Marcu, and Thomas Morton for comments on earlier versions of this work. We would also like to thank Slava Andreyev and Chi Lu for programming assistance.

References

[1] Allan, J. Carbonell, J., Doddington, G., Yamron, J. and Yang,Y. "Topic Detection and Tracking Pilot Study: Final Report." Proceedings of the Broadcast News Transcription and Understanding Workshop, pp. 194-218, 1998.

[2] Attali, Y., & Burstein, J. (2004, June). Automated essay scoring with e-rater V.2.0. To be presented at the Annual Meeting of the International Association for Educational Assessment, Philadelphia, PA.

[3] Billsus, D. & Pazzani, M. (1999). A Hybrid User Model for News Story Classification, Proceedings of the Seventh International Conference on User Modeling (UM '99), Banff, Canada, June 20-24, 1999.

[4] Burstein, J. et al. (1998). Automated Scoring Using A Hybrid Feature Identification Technique. Proceedings of 36th Annual Meeting of the Association for Computational Linguistics, 206-210. Montreal, Canada

[5] Burstein, J. et al (2004). Automated essay evaluation: The Criterion online writing service. AI Magazine, 25(3), 27-36.

[6] Burstein, J. (2003) The e-rater® scoring engine: Automated essay scoring with natural language processing. In Anonymous (Eds.), Automated essay scoring: A cross-disciplinary perspective. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

[7] Chapman WW, Christensen LM, Wagner MM, Haug PJ, Ivanov O, Dowling JN, Olszewski RT. (in press). Classifying Free-text Triage Chief Complaints into Syndromic Categories with Natural Language Processing. Artificial Intelligence in Medicine.

[8] Cohen, William W., Carvalho Vitor R., & Mitchell, Tom (2004): Learning to Classify Email into "Speech Acts" in EMNLP 2004.

[9] Elliott, S. 2003. Intellimetric: From Here to Validity. In Shermis, M., and Burstein, J. eds. Automated essay scoring: A cross-disciplinary perspective. Hillsdale, NJ: Lawrence Erlbaum Associates.

[10] Foltz, P. W., Kintsch, W., and Landauer, T. K. 1998. Analysis of Text Coherence Using Latent Semantic Analysis. Discourse Processes 25(2-3):285-307.

[11] Harman, Donna. 1992. The DARPA TIPSTER project. SIGIR Forum 26(2), 26-28.

[12] Hripcsak, G., Friedman, C., Alderson, P. O., DuMouchel, W., Johnson, S. B. and Clayton, P. D. (1995). Unlocking Clinical Data from Narrative Reports: A Study of Natural Language Processing Ann Intern Med, 122(9): 681 - 688.

[13] Joachims, T. (2002). Optimizing Search Engines Using Clickthrough Data, Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD).

[14] Larkey, L. 1998. Automatic Essay Grading Using Text Categorization Techniques. Proceedings of the 21st ACM-SIGIR Conference on Research and Development in Information Retrieval, 90-95. Melbourne, Australia.

[15] McCallum, Andrew, Nigam, Kamal, Rennie, Jason and Seymore, Kristie. Building Domain-Specific Search Engines with Machine Learning Techniques. AAAI-99 Spring Symposium.

[16] Page, E. B. 1966. The Imminence of Grading Essays by Computer. Phi Delta Kappan, 48:238-243.

[17] Sahami, M., Dumais, S., Heckerman, D., and Horvitz, E. 1998. A Bayesian Approach to Filtering Junk E-Mail. In Learning for Text Categorization: Papers from the 1998 Workshop. AAAI Technical Report WS-98-05.

[18] Sahlgren, Magnus. 2001. Vector-based semantic analysis: Representing word meanings based on random labels. In Proceedings of the ESSLLI 2001 Workshop on Semantic Knowledge Acquisition and Categorisation. Helsinki, Finland.

[19] Salton, Gerard. 1989. Information Retrieval: Data Structures and Algorithms. Reading, Massachussetts: Addison-Wesley.

[20] Wilcox AB, Hripcsak G. The role of domain knowledge in automating medical text report classification. J Am Med Inform Assoc 2003;10:330–8.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Thread-based analysis of patterns of collaborative interaction in chat

Murat Cakir, Fatos Xhafa, Nan Zhou and Gerry Stahl
Drexel University
Philadelphia, PA 19104, USA

Abstract. In this work we present a thread-based approach for analyzing synchronous collaborative math problem solving activities. Thread information is shown to be an important resource for analyzing collaborative activities, especially for conducting sequential analysis of interaction among participants of a small group. We propose a computational model based on thread information which allows us to identify patterns of interaction and their sequential organization in computer-supported collaborative environments. This approach enables us to understand important features of collaborative math problem solving in a chat environment and to envisage several useful implications for educational and design purposes.

1. Introduction

The analysis of fine-grained patterns of interaction in small groups is important for understanding collaborative learning [1]. In distance education, collaborative learning is generally supported by asynchronous threaded discussion forums and by synchronous chat rooms. Techniques of interaction analysis can be borrowed from the science of conversation analysis (CA), adapting it for the differences between face-to-face conversation and online discussion or chat. CA has emphasized the centrality of turn-taking conventions and of the use of adjacency pairs (such as question-answer or offer-response interaction patterns). In informal conversation, a given posting normally responds to the previous posting. In threaded discussion, the response relationships are made explicit by a note poster, and are displayed graphically. The situation in chat is more complicated, and tends to create confusions for both participants and analysts.

In this paper, we present a simple mathematical model of possible response structures in chat, discuss a program for representing those structures graphically and for manipulating them, and enumerate several insights into the structure of chat interactions that are facilitated by this model and tool. In particular, we show that fine-grained patterns of collaborative interaction in chat can be revealed through statistical analysis of the output from our tool. These patterns are related to social, communicative and problem-solving interactions that are fundamental to collaborative learning group behavior.

Computer-Supported Collaborative Learning (CSCL) research has mainly focused on analyzing content information. Earlier efforts aimed at identifying interaction patterns in chat environments such as Soller et al. [2] were based on the ordering of postings generated by the system. A naïve sequential analysis solely based on the observed ordering of postings without any claim about their threading might be misleading due to artificial turn orderings produced by the quasi-synchronous chat medium [3], particularly in groups larger than two or three [4].

In recent years, we have seen increasing attention on thread information, yet most of this research is focused on asynchronous settings ([5], [6], [7], [8], [9]). Jeong [10] and Kanselaar et al. [11], for instance, use sequential analysis to examine group interaction in asynchronous threaded discussion. In order to do a similar analysis of chat logs, one has to first take into account the more complex linking structures.

Our approach makes use of the thread information of the collaboration session to construct a graph that represents the flow of interaction, with each node denoting the content that includes the complete information from the recorded transcript. By traversing the graph, we mine the most frequently occurring dyad and triad structures, which are analyzed more closely to identify the patterns of collaboration and sequential organization of interaction under such specific setting. The proposed thread-based sequential analysis is robust and scalable, and thus can be applied to study synchronous or asynchronous collaboration in different contexts.

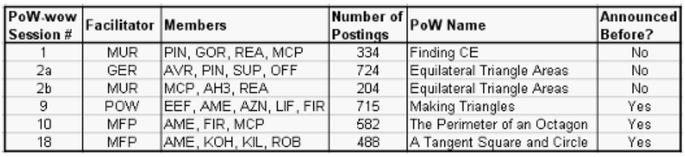
The rest of the paper is organized as follows: Section 2 introduces the context of the research, including a brief introduction of the Virtual Math Teams project, and the coding scheme on which the thread-based sequential analysis is based. Section 3 states the research questions we want to investigate. In Section 4 we introduce our approach. We present interesting findings and discuss them to address our research questions and to envisage several useful implications for educational and design purposes in Section 5. Section 6 concludes this work and points to future research.

2. Context of the Research

The VMT Project and Data Collection

The Virtual Math Teams (VMT) project at Drexel University investigates small group collaborative learning in mathematics. In this project an experiment is being conducted, called *powwow*, which extends The Math Forum's (mathforum.org) "*Problem of the Week (PoW)*" service. Groups of 3 to 5 students in grades 6 to 11 collaborate online synchronously to solve math problems that require reflection and discussion. AOL's Instant Messenger software is used to conduct the experiment in which each group is assigned to a chat room. Each session lasts about one to one and a half hour. The *powwow* sessions are recorded as chat logs (transcripts) with the handle name (the participant who made the posting), timestamp of the posting, and the content posted (see Table 1). The analysis conducted in this paper is based on 6 of these sessions. In 3 of the 6 sessions the math problem was announced at the beginning of the session, whereas in the rest the problem was posted on the Math Forum's web site in advance.

Table 1: Description of the coded chat logs.

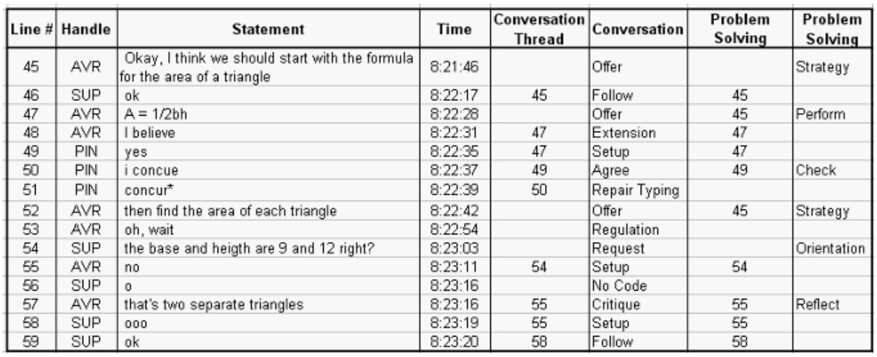


PoW-wow Session #	Facilitator	Members	Number of Postings	PoW Name	Announced Before?
1	MUR	PIN, GOR, REA, MCP	334	Finding CE	No
2a	GER	AVR, PIN, SUP, OFF	724	Equilateral Triangle Areas	No
2b	MUR	MCP, AH3, REA	204	Equilateral Triangle Areas	No
9	POW	EEF, AME, AZN, LIF, FIR	715	Making Triangles	Yes
10	MFP	AME, FIR, MCP	582	The Perimeter of an Octagon	Yes
18	MFP	AME, KOH, KIL, ROB	488	A Tangent Square and Circle	Yes

Coding Scheme

Both quantitative and qualitative approaches are employed in the VMT project to analyze the transcripts in order to understand the interaction that takes place during collaboration within this particular setting. A coding scheme has been developed in the VMT project to quantitatively analyze the sequential organization of interactions recorded in a chat log. The unit of analysis is defined as one posting that is produced by a participant at a certain point of time and displayed as a single posting in the transcript.

The coding scheme includes nine distinct dimensions, each of which is designed to capture a certain type of information from a different perspective. They can be grouped into two main categories: one is to capture the content of the session whereas another is to keep track of the threading of the discussion, that is, how the postings are linked together. Among the content-based dimensions, conversation and problem solving are two of the most important ones which code the conversational and problem solving content of the postings. Related to these two dimensions are the Conversation Thread and the Problem Solving Thread, which provide the linking between postings, and thus introduce the relational structure of the data. The conversation thread also links fragmented sentences that span multiple postings. The problem solving thread aims to capture the relationship between postings that relate to each other by means of their mathematical content or problem solving moves (see Figure 1).



Line #	Handle	Statement	Time	Conversation Thread	Conversation	Problem Solving	Problem Solving
45	AVR	Okay, I think we should start with the formula for the area of a triangle	8:21:46		Offer		Strategy
46	SUP	ok	8:22:17	45	Follow	45	
47	AVR	A = 1/2bh	8:22:28		Offer	45	Perform
48	AVR	I believe	8:22:31	47	Extension	47	
49	PIN	yes	8:22:35	47	Setup	47	
50	PIN	i concue	8:22:37	49	Agree	49	Check
51	PIN	concur*	8:22:39	50	Repair Typing		
52	AVR	then find the area of each triangle	8:22:42		Offer	45	Strategy
53	AVR	oh, wait	8:22:54		Regulation		
54	SUP	the base and heigth are 9 and 12 right?	8:23:03		Request		Orientation
55	AVR	no	8:23:11	54	Setup	54	
56	SUP	o	8:23:16		No Code		
57	AVR	that's two separate triangles	8:23:16	55	Critique	55	Reflect
58	SUP	ooo	8:23:19	55	Setup	55	
59	SUP	ok	8:23:20	58	Follow	58	

Figure 1: A coded excerpt from Pow2a.

Each dimension has a number of subcategories. The coding is done manually by 3 trained coders independently after strict training assuring a satisfactory reliability. This paper is based on 4 dimensions only; namely the conversation thread, conversation dimension, problem solving thread, and problem solving dimension.

3. Research Questions

In this explorative study we will address the following research questions:

Research Question 1: What patterns of interaction are frequently observed in a synchronous, collaborative math problem solving environment?

Research Question 2: How can patterns of interaction be used to identify: (a) each member's level of participation; (b) the distribution of contributions among participants; and, (c) whether participants are organized into subgroups through the discussion?

Research Question 3: What are the most frequent patterns related to the main activities of the math problem solving? How do these patterns sequentially relate to each other?

Research Question 4: What are the (most frequent) minimal building blocks observed during "local" interaction? How are these local structures sequentially related together yielding larger interactional structures?

4. The Computational Model

We have developed software to analyze significant features of online chat logs. The logs must first be coded manually, to specify both the local threading connections and the content categories. When a spreadsheet file containing the coded transcript is given as input, the program generates two graph-based internal representations of the interaction, depending on the conversation and problem solving thread dimensions respectively. In this representation each posting is treated as a node object, containing a list of references pointing to other nodes according to the corresponding thread. Moreover, each node includes additional information about the corresponding posting, such as the original statement, the author of the posting, its timestamp, and the codes assigned in other dimensions. This representation makes it possible to study various different *sequential patterns*, where sequential means that postings involved in the pattern are linked according to the thread, either from the perspective of participants who are producing the postings or from the perspective of coded information.

After building a graph representation, the model performs traversals over these structures to identify frequently occurring sub-structures within each graph, where each sub-structure corresponds to a sequential pattern of interaction. Sequential patterns having different features in terms of their size, shape and configuration type are studied. In a generic format dyads of type C_i-C_j, and triads of type C_i-C_j-C_k where $i<j<k$ are examined in an effort to get information about the local organization of interaction. In this representation C_i stands for a variable that can be replaced by a code or author information. The ordering given by $i<j<k$ refers to the ordering of nodes by means of their relative positions in the transcript. It should be noted that a posting represented by C_j can only be linked to previous postings, say C_i where $i<j$. In this notation the size of a pattern refers to the number of nodes involved in the pattern (e.g. the size is 2 in the case of C_i-C_j). Initially the size is limited to dyads and triads since they are more likely to be observed in a chat environment involving 3 to 5 participants. Nonetheless, the model can capture patterns of arbitrary size whenever necessary. The shape of the pattern refers to the different combinations in which the nodes are related to each other. For instance, in the case of a triad like C_i-C_j-C_k there are two possible type configurations: (a) if C_i is linked to C_j and C_j is linked to C_k , then we refer to this structure as *chain* type; (b) if C_i is linked to C_j and C_i is linked to C_k, then we refer to this structure as *star* type. The dyadic and triadic patterns identified this way reveal information about the local organization of interaction. Thus, these patterns can be considered as the fundamental building blocks of a group's discussion, whose combination would give us further insights on the sequential unfolding of the whole interaction.

The type of the configuration is determined by the information represented by each variable C_i. A variable C_i can be replaced by the author name, the conversation code, the problem solving code, or a combination of conversation and problem solving codes. This flexibility makes it possible to analyze patterns linking postings by means of their authors, and the codes they receive from the conversational or problem solving dimension.

As shown in Table 1, the maximum number of chat lines contained in a transcript in our data repository is about 700 lines, and we analyzed a corpus containing 6 such transcripts for this explorative study. Thus, in this study the emphasis is given to ways of revealing relevant patterns of collaborative interaction from a given data set. Nonetheless, we take care of efficiency issues while performing the mining task. Moreover, there exist efficient algorithms designed for mining frequent substructures in large graphs ([12], [13], [14]), which can be used to extend our model to process larger data sets.

5. Results and Discussion

In this section we show how the computational model presented in this work enables us to shed light on the research questions listed in Section 3.

5.1 Local Interaction Patterns

In order to identify the most frequent local interaction patterns of size 2 and 3, our model performs traversals of corresponding lengths and counts the number of observed dyads and triads. The model can classify these patterns in terms of their contributors, in terms of conversation or problem solving codes, or by considering different combinations of these attributes (e.g. patterns of author-conversation pairs). The model outputs a dyad percentage matrix for each session in which the $(i,j)^{th}$ entry corresponds to the percentage that C_i is followed by C_j during that session. For example, a percentage matrix for dyads based on conversation codes is shown in Table 2. In addition to this, a row-based percentage matrix is computed to depict the local percentage of any dyad C_i-C_j among all dyads beginning with C_i.

Table 3 shows a row-based percentage matrix for the conversation dyads. Similarly, the model also computes a list of triads and their frequencies for each session.

5.2 Frequent Conversational Patterns

For the conversational dyads we observed that there are a significant number of zero-valued entries on all six percentage matrices. This fact indicates that there are strong causal relationships between certain pairs of conversation codes. For instance, the event that an *Agree* statement is followed by an *Offer* statement is very unlikely due to the fact that the *Agree-Offer* pair has a zero value in all 6 matrices. By the same token, non-zero valued entries corresponding to a pair C_i-C_j suggests which C_i variables are likely to produce a reply of some sort. Moreover, C_j variables indicate the most likely replies that a conversational action C_i will get. This motivated us to call the most frequent C_i-C_j pairs as *source-sink* pairs, where the source C_i most likely solicits the action C_j as the next immediate reply.

The most frequent conversational dyads in our sample turned out to be *Request-Response* (16%, 7%, 9%, 9%, 10%, 8% for the 6 powwows respectively), *Response-Response* (12%, 5%, 2%, 4%, 10%, 11%) and *State-Response* (8%, 6%, 4%, 2%, 5%, 16%) pairs. In our coding scheme conversational codes *State*, *Respond, Request* are assigned to those statements that belong to a general discussion, while codes such as *Offer, Elaboration, Follow, Agree, Critique* and *Explain* are assigned to statements that are specifically related to the problem solving task. Thus, the computations show that a significant portion of the conversation is devoted to topics that are not specifically about math problem solving. In addition to these, dyads of type *Setup-X* (8%, 14%, 12%, 2%, 3%, 4%) and *X-Extension* (14%, 15%, 9%, 7%, 9%, 6%) are also among the most frequent conversational dyads. In compliance with their definitions, *Setup* and *Extension* codes are used for linking fragmented statements of a single author that span multiple chat lines. In these cases the fragmented parts make sense only if they are considered together as a single statement. Thus, only one of the fragments is assigned a code revealing the conversational action of the whole statement, and the rest of the fragments are tied to that special fragment by using *Setup* and *Extension* codes. The high percentage of *Setup-X* and *X-Extension* dyads shows that some participants prefer to interact by posting fragmented statements during chat. The high percentage of fragmented statements strongly affects the distribution of other types of dyadic patterns. Therefore, a "pruning" option is included in our model to combine these fragmented statements into a single node to reveal other source-sink relationships.

5.3 Handle Patterns

Frequent dyadic and triadic patterns based on author information can be very informative for making assessments about each participant's level and type of participation. For instance, Table 4 contrasts two groups, namely Pow2a and Pow2b (hereafter, group A and B, resp.) that worked on the same math problem in terms of their author-dyad percentages. In both matrices an entry *(i,j)* corresponds to the percentage of the event that the postings of participant *i* were conversationally related to the postings of participant *j* during the session. For the non-pruned matrices, entries on the diagonal show us the percentage that the same participant either extended or elaborated his/her own statement. For the pruned matrices the "noise" introduced by the fragmented statements is reduced by considering them together as a single unit. In the pruned case diagonal entries correspond to elaboration statements following a statement of the same participant.

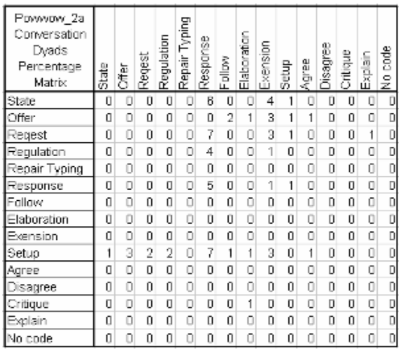
The most striking difference between the two groups, after pruning, is the difference between the percentage values on the diagonal: 10% for group A and 30% for group B. The percentages of most frequent triad patterns[1] show a similar behavior. The percentage of triads having the same author on all 3 nodes (e.g. AVR-AVR-AVR) is 15% for group A, and 42% for group B.

[1] For more results and our coding scheme refer to http://mathforum.org/wiki/VMT?ThreadAnalResults.

The pattern we see in group B is called an elaboration, where a member takes an extended turn. The pattern in group A indicates group exploration where the members collaborate to co-construct knowledge and turns rarely extend over multiple pruned nodes.

Patterns that contain the same author name on all its nodes are important indicators of individual activity, which typically occurs when a group member sends repeated postings without referring to any other group member. We call this elaboration, where one member of the group explains his/her ideas The high percentage of these patterns can be considered as a sign of separate threads in ongoing discussion, which is the case for group B. Moreover, there is an anti-symmetry between MCP's responses to REA's comments (23%) versus REA's responses to MCP's comments (14%). This shows that REA attended less to MCP's comments, than MCP to REA's messages. In contrast, we observe a more balanced behavior in group A, especially between AVR-PIN (17%, 18%) and AVR-SUP (13%, 13%). Another interesting pattern for group A is that the balance with respect to AVR does not exist between the pair SUP-PIN. This suggests that AVR was the dominant figure in group A, who frequently attended to the other two members of the group. To sum up, this kind of analysis points out similar results concerning roles and prominent actors as addressed by other social network analysis techniques.

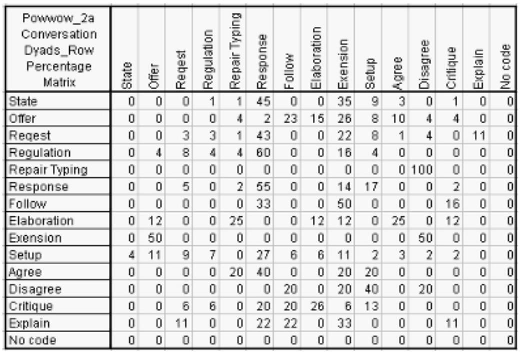
Table 2: Conversation dyads



Powwow_2a Conversation Dyads Percentage Matrix	State	Offer	Reqest	Regulation	Repair Typing	Response	Follow	Elaboration	Exension	Setup	Agree	Disagree	Critique	Explain	No code
State	0	0	0	0	0	6	0	0	4	1	0	0	0	0	0
Offer	0	0	0	0	0	0	2	1	3	1	1	0	0	0	0
Reqest	0	0	0	0	0	7	0	0	3	1	0	0	0	1	0
Regulation	0	0	0	0	0	4	0	0	1	0	0	0	0	0	0
Repair Typing	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Response	0	0	0	0	0	5	0	0	1	1	0	0	0	0	0
Follow	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Elaboration	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Exension	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Setup	1	3	2	2	0	7	1	1	3	0	1	0	0	0	0
Agree	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Disagree	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Critique	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
Explain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
No code	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The %s are computed over all pairs

Table 3: Row based distribution of conversation dyads



Powwow_2a Conversation Dyads_Row Percentage Matrix	State	Offer	Reqest	Regulation	Repair Typing	Response	Follow	Elaboration	Exension	Setup	Agree	Disagree	Critique	Explain	No code
State	0	0	0	1	1	45	0	0	35	9	3	0	1	0	0
Offer	0	0	0	0	4	2	23	15	26	8	10	4	4	0	0
Reqest	0	0	3	3	1	43	0	0	22	8	1	4	0	11	0
Regulation	0	4	8	4	4	60	0	0	16	4	0	0	0	0	0
Repair Typing	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
Response	0	0	5	0	2	55	0	0	14	17	0	0	2	0	0
Follow	0	0	0	0	0	33	0	0	50	0	0	0	16	0	0
Elaboration	0	12	0	0	25	0	0	12	12	0	25	0	12	0	0
Exension	0	50	0	0	0	0	0	0	0	0	0	50	0	0	0
Setup	4	11	9	7	0	27	6	6	11	2	3	2	2	0	0
Agree	0	0	0	0	20	40	0	0	20	20	0	0	0	0	0
Disagree	0	0	0	0	0	0	20	0	20	40	0	20	0	0	0
Critique	0	0	6	6	0	20	20	26	6	13	0	0	0	0	0
Explain	0	0	11	0	0	22	22	0	33	0	0	0	11	0	0
No code	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The %s are computed separately for each row

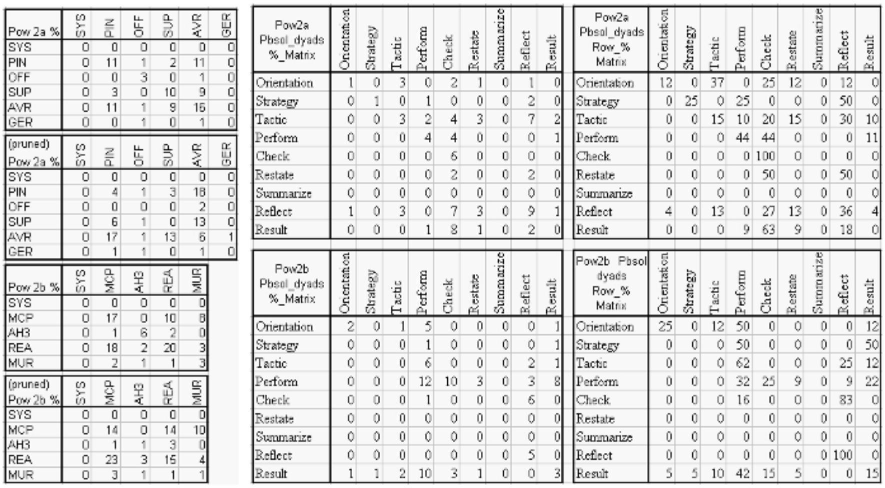
Dyadic and triadic patterns can also be useful in determining which member was most influential in initiating discussion during the session. For a participant *i,* the sum of row percentages *(i,j)* where $i \neq j$ can be used as a metric to see who had more initiative as compared to other members. The metric can be improved further by considering the percent of triads initiated by user *i*. For instance, in group A the row percentages are 31%, 22%, 20% and 2% for AVR, PIN, SUP and OFF respectively and the percentage of triads initiated by each of them is 41%, 29%, 20% and 7%. These numbers show that AVR had a significant impact in initiating conversation. In addition to this, a similar metric for the columns can be considered for measuring the level of attention a participant exhibited by posting follow up messages to other group members.

5.4 Problem Solving Patterns

A similar analysis of dyadic and triadic patterns can be used for making assessments about the local organization of a group's problem solving actions. The problem solving data produced by our model for groups A and B will be used to aid the following discussion in this section. Table 4 displays both groups' percentage matrices for problem solving dyads.

Before making any comparisons between these groups, we briefly introduce how the coding categories are related to math problem solving *activities*. In this context a problem solving activity refers to a set of successive math problem solving actions. In our coding scheme, *Orientation*, *Tactic* and *Strategy* codes refer to the elements of a certain activity in which the group engages in understanding the problem statement and/or proposes strategies for approaching it. Next, a combination of *Perform* and *Result* codes signal actions that relate to an execution activity in which previously proposed ideas are applied to the problem. *Summary* and *Restate* codes arise when the group is in the process of helping a group member to catch up with the rest of the group and/or producing a reformulation of the problem at hand. Further, *Check* and *Reflect* codes capture moves where group members reflect on the validity of an overall strategy or on the correctness of a specific calculation; they do not form an activity by themselves, but are interposed among the activities described before

Table 4: Handle & Problem Solving Dyads for Pow2a and Pow2b



Pow 2a %	SYS	PIN	OFF	SUP	AVR	GER
SYS	0	0	0	0	0	0
PIN	0	11	1	2	11	0
OFF	0	0	3	0	1	0
SUP	0	3	0	10	9	0
AVR	0	11	1	9	16	0
GER	0	0	1	0	1	0

(pruned) Pow 2a %	SYS	PIN	OFF	SUP	AVR	GER
SYS	0	0	0	0	0	0
PIN	0	4	1	3	18	0
OFF	0	0	0	0	2	0
SUP	0	6	1	0	13	0
AVR	0	17	1	13	6	1
GER	0	1	1	0	1	0

Pow 2b %	SYS	MCP	AH3	REA	MUR
SYS	0	0	0	0	0
MCP	0	17	0	10	8
AH3	0	1	6	2	0
REA	0	18	2	20	3
MUR	0	2	1	1	3

(pruned) Pow 2b %	SYS	MCP	AH3	REA	MUR
SYS	0	0	0	0	0
MCP	0	14	0	14	10
AH3	0	1	1	3	0
REA	0	23	3	15	4
MUR	0	3	1	1	1

Pow2a Pbsol_dyads %_Matrix	Orientation	Strategy	Tactic	Perform	Check	Restate	Summarize	Reflect	Result
Orientation	1	0	3	0	2	1	0	1	0
Strategy	0	1	0	1	0	0	0	2	0
Tactic	0	0	3	2	4	3	0	7	2
Perform	0	0	0	4	4	0	0	0	1
Check	0	0	0	0	6	0	0	0	0
Restate	0	0	0	0	2	0	0	2	0
Summarize	0	0	0	0	0	0	0	0	0
Reflect	1	0	3	0	7	3	0	9	1
Result	0	0	0	1	8	1	0	2	0

Pow2a Pbsol_dyads Row_% Matrix	Orientation	Strategy	Tactic	Perform	Check	Restate	Summarize	Reflect	Result
Orientation	12	0	37	0	25	12	0	12	0
Strategy	0	25	0	25	0	0	0	50	0
Tactic	0	0	15	10	20	15	0	30	10
Perform	0	0	0	44	44	0	0	0	11
Check	0	0	0	0	100	0	0	0	0
Restate	0	0	0	0	50	0	0	50	0
Summarize	0	0	0	0	0	0	0	0	0
Reflect	4	0	13	0	27	13	0	36	4
Result	0	0	0	9	63	9	0	18	0

Pow2b Pbsol_dyads %_Matrix	Orientation	Strategy	Tactic	Perform	Check	Restate	Summarize	Reflect	Result
Orientation	2	0	1	5	0	0	0	0	1
Strategy	0	0	0	1	0	0	0	0	1
Tactic	0	0	0	6	0	0	0	2	1
Perform	0	0	0	12	10	3	0	3	8
Check	0	0	0	1	0	0	0	6	0
Restate	0	0	0	0	0	0	0	0	0
Summarize	0	0	0	0	0	0	0	0	0
Reflect	0	0	0	0	0	0	0	5	0
Result	1	1	2	10	3	1	0	0	3

Pow2b Pbsol dyads Row_% Matrix	Orientation	Strategy	Tactic	Perform	Check	Restate	Summarize	Reflect	Result
Orientation	25	0	12	50	0	0	0	0	12
Strategy	0	0	0	50	0	0	0	0	50
Tactic	0	0	0	62	0	0	0	25	12
Perform	0	0	0	32	25	9	0	9	22
Check	0	0	0	16	0	0	0	83	0
Restate	0	0	0	0	0	0	0	0	0
Summarize	0	0	0	0	0	0	0	0	0
Reflect	0	0	0	0	0	0	0	100	0
Result	5	5	10	42	15	5	0	0	15

SYS refers to system messages. GER and MUR are facilitators of the groups.

Given this description, we use the percentage matrices (see Table 4) to identify what percent of the overall problem solving effort is devoted to each activity. For instance, the sum of percentage values of the sub-matrix induced by the columns and rows of *Orientation, Tactic, Strategy, Check* and *Reflect* codes takes up 28% of the problem solving actions performed by the group A, whereas this value is only 5% for group B. This indicates that group A put more effort in developing strategies for solving the problem. When we consider the sub-matrix induced by *Perform, Result, Check* and *Reflect*, the corresponding values are 21% for group A and 50% for group B. This signals that group B spent more time on executing problem solving steps. Finally, the values of the corresponding sub-matrix induced by *Restate, Summarize, Check*, and *Reflect* codes adds up to 7% for group A and 0% for B, which hints at a change in orientation of group A's problem solving activity. The remaining percentage values excluded by the sub-matrices belong to transition actions in between different activities.

5.5 Maximal Patterns

The percentage values presented in the previous section indicate that groups A and B exhibited significantly different local organizations in terms of their problem solving activities. In order to make stronger claims about the differences at a global level one needs to consider the unfolding of these local events through the whole discussion. Thus, analyzing the sequential unfolding of local patterns is another interesting focus of investigation which will ultimately yield a "global" picture of a group's collaborative problem solving activity. For instance, given the operational descriptions of problem solving activities in Subsection 5.4, we observed the following sequence of local patterns in group A. First, the group engaged in a problem orientation activity in which they identified a relevant sub-problem to work on. Then, they performed an execution activity on the agreed strategy by making numerical calculations to solve their sub-problem. Following this discussion, they engaged in a reflective activity in which they tried to relate the solution of the sub-problem to the general problem. During their reflection they realized they made a mistake in a formula they used earlier. At that point the session ended, and the group failed to produce the correct answer to their problem. On the other hand, the members of group B individually solved the problem at the beginning of the session without specifying a group strategy. They spent most of the remaining discussion revealing their solution steps to each other.

6. Conclusion and Ongoing Research

In this work we have shown how thread information can be used to identify the most frequent patterns of interaction with respect to various different criteria. In particular, we have discussed how these patterns can be used for making assessments about the organization of interaction in terms of each participant's level of participation, the conversational structure of discussion as well as the problem solving activities performed by the group. Our computations are based on an automated program which accepts a coded chat transcript as input, and performs all necessary computations in an efficient way.

In our ongoing research we are studying other factors that could influence the type of the patterns and their frequencies, such as the group size, the type of the math problem under discussion, etc. Moreover, we are investigating whether the interaction patterns and the problem solving phases reveal information about the type of the organization of the interaction, e.g. exploratory vs. reporting work. Finally, we will be using our data to feed a statistical model and thus study the research questions from a statistical perspective. We are also planning to extend the existing computational model to support XML input in order to make the model independent of the specific features introduced by a coding scheme.

References

[1] Stahl, G. (2006). Group Cognition: Computer Support for Building Collaborative Knowledge. *Cambridge, MA: MIT Press*.

[2] Soller, A., and Lesgold, A. (2003) A computational approach to analyzing online knowledge sharing interaction. *Proceedings of AI in Education 2003, Sydney, Australia, 253-260.*

[3] Garcia, A., and Jacobs, J.B. (1998). The interactional organization of computer mediated communication in the college classroom. *Qualitative Sociology, 21(3), 299-317.*

[4] O'Neil, J., and Martin, D. (2003). Text chat in action. *Proceedings of the international ACM SIGGROUP conference on Supporting group work, Sanibel Island, Florida, USA, 40-49.*

[5] Smith, M., Cadiz, J., and Burkhalter, B. (2000) Conversation Trees and Threaded Chats, *Proceedings of the 2000 ACM conference on Computer supported cooperative work, Philadelphia, PA, USA, 97-105.*

[6] Popolov, D., Callaghan, M., and Luker, P. (2000). Conversation Space: Visualising Multi-threaded Conversation. *Proceedings of the working conference on Advanced visual interfaces, Palermo,Italy,246-249*

[7] King, F.B., and Mayall, H.J. (2001) Asynchronous Distributed Problem-based Learning, *Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT'01), 157-159.*

[8] Tay, M.H., Hooi, C.M., and Chee, Y.S. (2002) Discourse-based Learning using a Multimedia Discussion Forum. *Proceedings of the International Conference on Computers in Education (ICCE'02), IEEE, 293.*

[9] Venolia, G.D. and Neustaedter, C. (2003) Understanding Sequence and Reply Relationships within Email Conversations: A mixed-model visualization. *Proceedings of SIGCHI'03, Ft. Lauderdale, FL, USA,361-368.*

[10] Jeong, A.C. (2003). The Sequential Analysis of Group Interaction and Critical Thinking in Online Threaded Discussion. *The American Journal of Distance Education, 17(1), 25-43.*

[11] Kanselaar, G., Erkens, G., Andriessen, J., Prangsma, M., Veerman, A., and Jaspers, J. (2003) Designing Argumentation Tools for Collaborative Learning. Book chapter of *Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-Making, Kirschner, P.A., et al. eds, Springer.*

[12] Inokuchi, A., Washio, T. and Motodam H. (2000). An apriori-based algorithm for mining frequent substructures from graph data. *Proceedings of PKDD 2000, Lyon, France, 13-23.*

[13] Kuramochi,M. and Karypis, G. (2001). Frequent subgraph discovery. *Proceedings of the 2001 IEEE International Conference on Data Mining, San Jose, California, USA, 313-320.*

[14] Zaki, M.J. (2002). Efficiently mining frequent trees in a forest. *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, Edmonton, Canada, 71-80.*

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Conceptual Conflict by Design: Dealing with Students' Learning Impasses in Multi-user Multi-agent Virtual Worlds

Yam San CHEE & Yi LIU
National Institute of Education, Nanyang Technological University
1 Nanyang Walk, Singapore 637616

Abstract. This paper describes current work directed at dealing with students' learning impasses that can arise when they are unable to make further learning progress while interacting in a 3D virtual world. This kind of situation may occur when group members do not possess the requisite knowledge needed to bootstrap themselves out of their predicament or when all group members mistakenly believe that their incorrect conceptual understanding of a science phenomenon is correct. The work reported here takes place in C–VISions, a socialized collaborative learning environment. To deal with such learning impasses, we have developed multiple embodied pedagogical agents and introduced them into the C–VISions environment. The agents are used to trigger experientially grounded cognitive dissonance between students and thereby to induce conceptual conflict that requires resolution. We describe the design and implementation of our agents which take on different functional roles and are programmed to aid students in the conflict resolution process. A description of multi agent-user interaction is provided to demonstrate how the agents enact their roles when students encounter a learning impasse.

1. Introduction

C–VISions [1] is a multi-user 3D virtual world environment for collaborative learning based on virtual interactive simulations. The pedagogical foundations of the learning approach adopted here can be found in Kolb's [2] Experiential Learning Cycle. Kolb's theory recognizes the vital role of experiential grounding in individual sense making and knowledge construction. The argument is that active experimentation on the part of learners gives rise to concrete experience which, in turn, provides the basis for reflective observation that can then lead to abstract conceptualization. Concepts so formed can then be tested by further active experimentation and so on, with the cycle repeating. This theory provides an authentic account of how humans acquire concepts and an understanding of phenomena that is intrinsically meaningful. In so doing, it overcomes the problems of symbol grounding [3] and semantic bootstrapping [4] that plague traditional accounts of human cognition.

In an earlier pilot study [5], we reported how three students using the C–VISions environment to learn about Newtonian physics would sometimes find themselves in a dilemma when the behavior that they hypothesized would occur did not materialize and they also realized that the explanation they could provide for the unexpected observation was unpersuasive. Sometimes, the explanation they provided was actually incorrect, and it might have been based on a shared misconception. These situations sensitized us to the collaborative learning impasses that can arise in peer-to-peer learning when the knowledge of students is homogenous, perhaps because they have the same schoolteacher. Hence, we have tried to address this problem by introducing pedagogical agents as a means of helping students to bootstrap themselves out of their collaborative learning impasses.

In the next section of this paper, we provide some of the background to research in the field of pedagogical agents in virtual environments. We then explain the learning design of our virtual environment and how we have designed our agents, elaborating on the agent architecture and its implementation. We next describe an extended interaction episode between users and agents to demonstrate the nature of learning interaction that occurred based on a set of agent heuristics that we have framed. Finally, we conclude the paper, highlighting challenges related to future work.

2. Research background

The integration of the agent technology with learning simulation systems can enhance student learning by providing interactive guidance in a natural and rich way. Humanlike agents are usually constructed as domain experts to help users overcome learning difficulties and present just-in-time knowledge. One of the most well-known pedagogical agents is Steve, an embodied agent developed by Rickel and Johnson [6]. It acts as a virtual instructor to teach students how to maneuver a submarine through demonstrating operations, monitoring student behaviors, and giving clear explanations. A hierarchical approach is used to define Steve's task procedures. Steps are defined as nodes while the causal relations between steps are represented as links. Ordering constraints allow Steve to present information in a logically ordered sequence. Causal links also identify the pre and post conditions of each task step. WhizLow [7] is another 3D agent. It inhabits a virtual CPU City and explains concepts about computer architecture to students who navigate along different virtual computer components. The agent's responses are triggered in response to different user's misconceptions that are detected. Herman [8], yet another virtual embodied agent, helps students to learn biology by allowing them to customize a virtual plant and to foster its growth. Herman has been designed as a reactive agent that interrupts students' actions as soon as they perform an inappropriate step.

The examples of agents cited above are all instances of systems that contain only one agent. Multi-agent systems allow a team of agents to interact with one or more users simultaneously. However, the design and implementation of such systems present significant challenges because of the requirement to also model multiparty interaction in the virtual environment. The Mission Rehearsal Exercise project [9] contains an interactive peacekeeping scenario with sergeant, mother, and medic in the foreground. A set of interaction layers for multiparty interaction control regarding contact, attention, conversation, social commitments, and negotiation are defined. In the conversation layer, components such as participants, turn, initiative, grounding, topic, and rhetoric are defined to build the computational model for social interaction and to facilitate the management of multiparty dialog. There has been little work on multi-user, multi-agent systems oriented toward supporting learning. Dignum & Vreeswijk [10] put forward various considerations for implementing multiparty interaction, including the idea of defining group interaction patterns. This concept of interaction patterns is further elaborated on by Suh [11] who proposes a taxonomy of interaction patterns for a tutoring scenario.

3. Learning Design

The design of learning tasks and processes in C–VISions adheres to the fundamental principle of grounding concept in percept [12]. Within this framework, we have adopted the approach of conflict resolution by design to help students bootstrap themselves out of learning impasses. Embodied agents, that we have introduced into the virtual world, are used to deliberately engender a sense of experientially grounded cognitive dissonance between students. By explicitly projecting experienced contradictions and conflicting understandings into the open, students are impelled to search for coherent causal explanations that resolve the apparent contradictions and discrepancies.

In our learning design, we employ multiple embodied agents with different functional roles. One agent specializes in giving help in the form of instructions, another specializes in evaluating what students say and do, and the third agent specializes in assisting students with conceptual and higher-order thinking. This division of labor between agents was inspired by the work of White & Frederiksen [13] which introduced a set of agents to support the process of students engaging in scientific inquiry. However, the agents of White & Frederiksen are passive; that is, they are channels to access useful information of different types, and they constitute an enhanced version of a software's help function. In contrast, our agents are intelligent agents that know how to interact with students as well as with one another. We are unaware of any prior work in the domain of learning technologies that supports multiple pedagogical agents interacting with multiple collaborating users.

4. Agent Design and Heuristics

In order to evolve the C–VISions system from a multi-user system to a multi-user multi-agent system, we introduced an agent architecture described in [14]. Figure 1 shows the schematic depiction of the architecture which comprises four layers: the proposition layer, the understanding layer, the expertise layer, and the reflexive layer. Multi-agent multi-user systems must provide some mechanism to enable sensible turn taking in conversational dialog between members of the heterogeneous group comprising humans, represented by avatars, and the embodied agents. Our approach to this problem is to make use of interaction models described in [15]. The agent architecture also maintains a shared user model for each user. The agents draw from these user models in determining their own behavior. A group dialog history is also maintained to help agents customize their responses to the evolving conversational context as it unfolds in real-time.

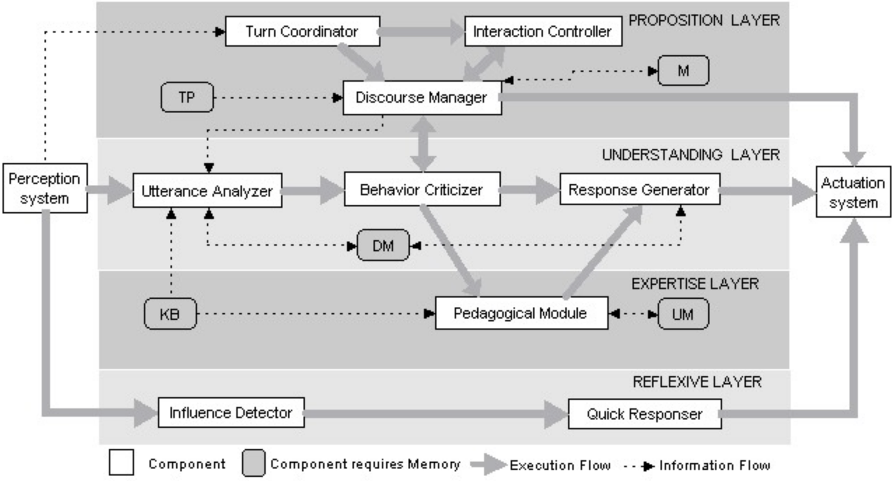


Figure 1: Four-layer agent architecture
[TP: Task Planner; M: Memory; DM: Dialog Model; KB: Knowledge Base; UM: User Model]

Within the space station and spaceship virtual world that we use to illustrate our work in this paper, there are three agents: Ivan, the Instructor agent, Ella, the Evaluator agent, and Tae, the conceptual Thinking agent. Each agent maintains a separate knowledge base that encodes the knowledge required by the agent to fulfill its functional role in relation to the students' learning task. Table 1 presents a sample of heuristics possessed by each agent that helps them collectively to facilitate collaborative and group learning behaviors. These heuristics are defined in terms of rules. The heuristics assume application of the principle of conceptual conflict by design. Application of the heuristics is annotated in Section 6.

Table 1. Agent heuristics based on role (sample only)

Ivan, the Instructor agent:		
IF detect that students lack a critical knowledge component	THEN provide information on the missing knowledge component	Rule 1
IF requested by Evaluator agent to set up a conceptual conflict	THEN choose an appropriate conflict task and provide the task information to students	Rule 2
IF students have just completed a task from different frames of reference	THEN invite users to share their different experiences (to project the conflict into the shared conceptual space for negotiation)	Rule 4
Ella, the Evaluator agent:		
IF detect that students have converged to a shared misconception	THEN request Instructor agent to set up a conceptual conflict for them to resolve	Rule 6
IF identify one student with a misconception and other students disagreeing	THEN ask other students to elaborate on the reasons for disagreeing	Rule 8
IF detect that students have made an error in constructing a model of the scientific phenomenon	THEN provide specific feedback on the step that is erroneous	Rule 11
Tae, the conceptual Thinking agent:		
IF conceptual conflict task has been set up by Instructor agent	THEN ask students to state a hypothesis or explanation that resolves the conflict	Rule 12
IF detect that students are not in agreement	THEN ask them to re-examine and reflect on what might be causing the disagreement	Rule 13
IF one student articulates his/her explanation	THEN ask another student for his/her opinion on it	Rule 14
IF two or more students answer a question	THEN ask the students whether their answers are in agreement	Rule 19

5. Supporting Multi Agent-User Interaction

The agents' heuristics serve as a foundation to support collaborative activity among multiple agents and users in the virtual environment. Two different types of collaboration can arise: conversational collaboration and learning task collaboration. Conversation level collaboration usually requires agents to understand the intention of users by interpreting their speech input. When an agent successfully interprets the shared intention of a group of users and agents, it infers a group level interaction pattern and uses it to generate suitable responses to deal with any interaction obstacles. For example, if several users constantly express opposing views, the group level interaction pattern is categorized as a disagreement situation. Hence, the agent will provide the necessary feedback to help users to identify the cause of the conflict. In contrast, remediation of task level collaboration usually takes place when users fail to carry out the learning activity in the required order. When this occurs, the agents have to reorganize their activities or modify their roles temporarily so as to construct a customized approach that will meet the specific learning requirements. For instance, if the Instructor agent invites two users to perform two distinct tasks as a preparation for their subsequent learning discussion and one of the users fails to act accordingly, the conceptual

Thinking agent will adopt this user's role temporarily and fulfill the required task. This arrangement provides the needed flexibility for the group interaction to proceed. Because of the distinct nature of these two types of collaboration, the implementation of the agent's heuristics is realized differently.

We make use of interaction patterns [11] to implement the agents' heuristics for conversational level collaboration. These patterns, usually extracted from real life tutoring situations, specify the basic turn taking information for multiparty learning scenarios. Each turn denotes an utterance or intention of either an agent or a user. Agents will always try to execute the inferred group pattern that applies to the situation.

In earlier work, we introduced the design of a task schema node to implement an agent's involvement in a user's task. This takes the form of a set of linked responses. Sequential links regulate the task flow while dialog links enable agents to trigger relevant feedback after processing the user's intention. When applying this schema approach in a multiparty environment, we additionally feature the schema node with a role and a precondition field. There are three benefits of doing so. First, the adoption of a role attribute extends the usage of the schema node to cover both agents' as well as users' behaviors in the virtual environment. As a result, agents gain the ability to analyze task collaboration taking into account the users' involvement. Second, the role field facilitates agents in identifying the appropriate action of a specific agent or user. Hence, whenever unexpected user behaviors arise, the agents can decide to take the responsibility for performing missing steps to preserve task flow. Third, the precondition schema field sets restrictions on the sequence of critical agents' and users' behaviors so as to help the agents maintain the logical order of steps for effective multiple agent-user interaction.

At the individual agent level, we implement an agent's understanding of what students say (by typing with a keyboard) in the following manner. First, a user's freeform natural language expression is parsed, using pattern matching, to yield a dialog act categorization [16]. Second, one or more relevant objects pertinent to the simulation domain (eg. car, spaceship) are identified by matching against an object keyword list. If more than one object is identified, the agent infers the most likely pertinent object of a student's expression based on dialog context. Using a knowledge base of object names, object attributes (eg. mass, horizontal velocity), properties of object attributes (eg. same, equal, change), and descriptors of actions on and changes to objects, the agent generates and ranks plausible states of a student's understanding. If necessary, this understanding can be translated from a predicate representation to a sentence, and the student can be requested to confirm whether the agent's inference of the student's understanding is correct. In this manner, an agent can construct a model of a student's evolving understanding.

6. Multi Agent-User Interaction Description

This section of the paper describes the setting from which the interaction protocol (removed due to lack of space) has been extracted. Two students, Jack and Mary, and the three agents, Ivan, Ella, and Tae, are in a virtual world designed to help students learn the concept of relative velocity (as well as other Newtonian physics concepts). The virtual world consists of a space station where the learning interaction takes place. A panel on the space station allows participants to control the movement of a spaceship that flies around the space station and to impose instantaneous amounts of force on the spaceship. There is also a four-wheeled utility vehicle that runs around on the space station platform (see Figure 2). The students have learned about the concept of relative velocity in school and have also read examples of relative velocity from their textbook. However, all the examples in their textbook involve motion in one direction only. These examples lead the students to subconsciously and incorrectly infer that relative velocity is a phenomenon that exists only in one-dimensional motion. The interaction description unfolds from this point.

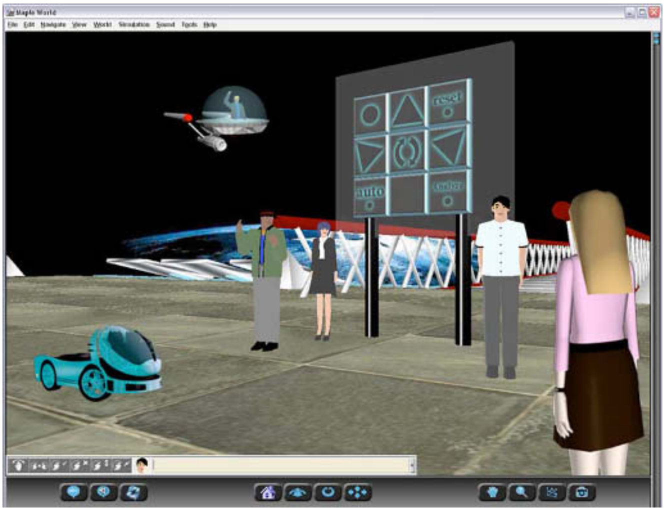


Figure 2: The virtual world setting with two students and three agents

Figure 3 depicts the situation, reflected in the interaction description below, when the agents help the students to understand that the concept of relative velocity also applies in two-dimensional motion. The agents do so by building a conceptual bridge from what the students experienced in the first person (using a dynamically generated replay of the motion that each student perceived) to a two-dimensional force diagram representation of the conflict that they are trying to resolve. The agent behaviors represent an attempt to scaffold student learning by providing a bridge between percept and concept.

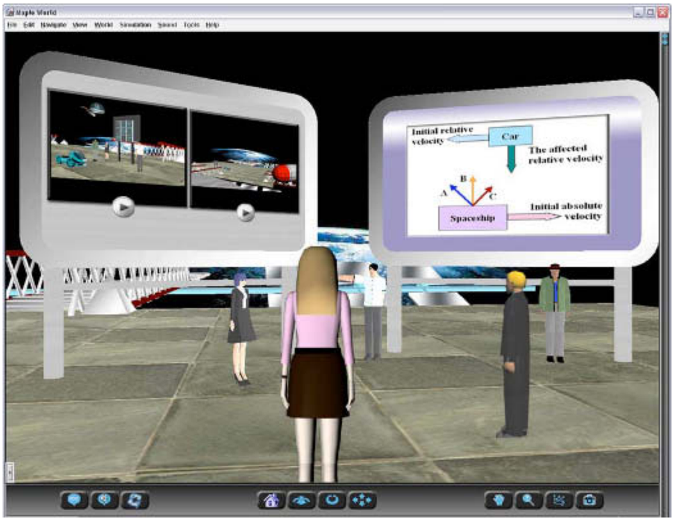

Maple World
File Edit Navigate View World Simulation Sound Tools Help
Initial relative velocity
Car
The affected relative velocity
B
A
C
Spaceship
Initial absolute velocity

Figure 3: Bridging from percept to concept in the domain of relative velocity

The transcript of the protocol proceeds as follows. The students Mary (in the foreground in Figures 2 and 3) and Jack (on the spaceship in Figure 2 and in the near foreground in Figure 3) are under the impression that the phenomenon of relative velocity only occurs in one-dimensional motion. The Evaluator agent, Ella, detects that the students share this misconception. She requests Ivan, the Instructor agent, to set up a conflict resolution situation to dislodge the students' misconception (Rule 6). Ivan asks Jack to teleport to a nearby spaceship and to observe the motion of a utility vehicle traveling along a straight path on the surface of the space station (Rule 2). The spaceship flies past at a low angle along a path parallel to the motion of the vehicle. Tae asks Jack what he expects the motion of the vehicle to look like from the spaceship (Rule 12). Meanwhile, Mary also watches the motion of the vehicle from the space station. Ivan then intentionally invites Mary to press one of the three directional arrows on the control panel to impose an instantaneous force on the spaceship, without Jack's knowledge. Mary presses the arrow in the left-most column of the second row of buttons. After the spaceship fly-past, Jack is teleported back to the space station. Ivan requests Jack and Mary to share their observations with one another (Rule 4). Mary reports seeing the vehicle moving along a straight course toward her. Jack reports seeing the vehicle moving in a direction opposite to the spaceship's direction. Mary and Jack are able to reconcile their dissimilar observations by appealing to the concept of relative velocity applied in one dimension. Tae asks them if their observations are in agreement after the application of the instantaneous force (Rule 19). However, Mary and Jack are unable to reconcile their mutual observations from the point when Jack experienced an unexpected instantaneous force on the spaceship.

To aid them in resolving this conflict, Tae, the conceptual Thinking agent (with arms raised in Figure 2) intervenes and invites Mary and Jack to compare videos of what they separately observed and to reflect on the differences (Rule 13). He directs their attention to the screen on the right and asks Jack to guess which button Mary pressed while he was on the spaceship. (These buttons correspond to the *direction* arrows A, B, and C on the screen. These arrows are not force vectors.) Jack makes a guess of direction C, but Mary interjects to say that she pressed the A direction arrow. Jack looks surprised. Tae, the thinking agent, asks Jack to explain why he thinks direction C is the correct answer (Rule 13). Jack states that this is how things appeared to him as the spaceship moved toward the space station. Tae asks Mary what she thinks of Jack's explanation (Rule 14). Mary answers that it cannot be correct and proceeds to explain, with reference to the diagram on the screen, that direction C is actually the *resultant* direction that arises from combining the spaceship's initial velocity and the force applied in direction A. Ella nods approvingly at Mary. However, Jack protests that, from what he observed, the car appeared to be moving *perpendicularly* toward him, with the side facing him; so he queries whether direction B should be the correct resultant direction instead. Tae asks Mary if she can resolve this dilemma for Jack. Mary shakes her head after pondering the request. At this point, Ella recognizes that Jack's observation of the car moving perpendicularly toward him is valid, and the spaceship moving in the resultant direction C is also valid because a very special situation has occurred: the amount of instantaneous force applied to the spaceship in direction A was such that it reduced the velocity of the spaceship to an amount exactly equal to the velocity of the car moving on the space station. To help the students recognize that this is a special case, Ella asks Ivan if he can set up another problem for the students to solve so that they would understand that what Jack observed was not a general case (Rule 6). So Ivan suggests that Jack and Mary re-perform the experiment. Unknown to both, Ivan increases the strength of the instantaneous force so that what Jack observes changes. This action leads to a fresh cycle of interaction between the students and the agents so that the students recognize the special characteristics of the earlier case. These cycles of interaction repeat until an equilibrium state of correct student conceptual understanding is achieved.

7. Conclusion

In this paper, we have outlined an approach to dealing with the problem of collaborative learning impasses that can arise when students engage in learning discourse and interaction in shared virtual world environments. We have implemented an approach, called conceptual conflict by design, where embodied pedagogical agents deliberately create situations of experiential conflict that triggers cognitive dissonance requiring resolution. In such environments, students can enjoy an enhanced sense of experiential involvement in learning-by-doing in the virtual world as well as a sense of immersion and co-presence with other social actors (both real and artificial), thereby helping learning to unfold in a natural, engaging, and humanistic way.

A key challenge of the system intelligence part of the development work revolves around dealing with the limitations of AI. Important issues that developers must address include defining and modeling the task structure of user-agent interaction, inferring the underlying user intentions and semantics without explicit probing, and programming agent decision making related to when to intervene and how to intervene. These problems are made somewhat more tractable by virtue of the fact that virtual worlds and learning task design effectively circumscribe the realm of meaningful and acceptable student actions.

References

[1] Chee, Y. S. & Hooi, C. M. (2002) C-VISions: Socialized learning through collaborative, virtual, interactive simultations. In *Proceedings of CSCL 2002*, pp. 687-696. Hillsdale, NJ: Lawrence Erlbaum.

[2] Kolb, D. A. (1984) *Experiential Learning: Experience as the Source of Learning and Development.* Englewood Cliffs, NJ: Prentice-Hall.

[3] Harnad, S. (1990) The symbol grounding problem. *Physica D*, *42*, 335–346.

[4] Edelman, G. (1992) *Bright Air, Brilliant Fire: On the Matter of the Mind.* Basic Books.

[5] Chee, Y. S. (2001) Networked virtual environments for collaborative learning. In *Proceedings of the Ninth International Conference on Computers in Education*, pp. 3-11.

[6] Rickel, J. & Johnson, W. L. (1998) STEVE: A pedagogical agent for virtual reality. In *Proceedings of the 2nd International Conference on Autonomous Agents*, pp. 332—333. ACM Press.

[7] Gregoire, J. P., Zettlemoyer, L. S., & Lester, J. C. (1999) Detecting and correcting misconceptions with lifelike avatars in 3D learning environments. In S. P. Lajoie & M. Vivet (Eds.) *Artificial Intelligence in Education: Open Learning Environments,* pp. 586-593. Amsterdam: IOS Press.

[8] Elliott, C., Rickel, J., & Lester, J. (1999) Lifelike pedagogical agents and affective computing: An exploratory synthesis. In M. Wooldridge & M. Veloso (Eds.), *Artificial Intelligence Today*, pp. 195-212. Springer-Verlag.

[9] Traum, D. & Rickel, J. (2002) Embodied agents for multi-party dialogue in immersive virtual worlds, In *Proceedings of the 2nd International Conference on Autonomous Agents and Multiagent Systems*, pp. 766-773.

[10] Dignum, F. P. M. & Vreeswijk, G. A. W. (2001) Towards a testbed for multi-party dialogues. In *AAMAS 2001 International Workshop on Agent Communication Languages and Conversation Policies*, pp. 63-71.

[11] Suh, H. J. (2001) A case study on identifying group interaction patterns of collaborative knowledge construction process. Paper presented at the 9th International Conference on Computers in Education.

[12] Chee, Y. S. & Liu, Y. (2004) Grounding concept in percept: Learning physics experientially in multi-user virtual worlds. In *Proceedings of ICALT 2004*, pp. 340-344. Los Alamitos, CA: IEEE Society.

[13] White, B. Y. & Frederiksen, J. R. (2000) Technological tools and instructional approaches for making scientific inquiry accessible to all. In M. J. Jacobson & R. B. Kozma (Eds.), *Innovations in Science and Mathematics Education*, pp. 321–359. Mahwah, NJ: Lawrence Erlbaum.

[14] Liu, Y. & Chee, Y. S. (2004) Intelligent pedagogical agents with multiparty interaction support. In *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pp. 134-140. Los Alamitos, CA: IEEE Computer Society.

[15] Liu, Y. & Chee, Y. S. (2004) Designing interaction models in a multiparty 3D learning environment. In *Proceedings of the Twelfth International Conference on Computers in Education*, pp. 293-302.

[16] S. I. Lee & S. B. Cho (2001). An intelligent agent with structured pattern matching for a virtual representative. In *Proceedings of the 2nd Asia-Pacific Conference on Intelligent Agent Technology,* pp. 305–302. World Scientific.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Motivating Learners by Nurturing Animal Companions: My-Pet and Our-Pet

Zhi-Hong Chen[1], Yi-Chan Deng[1], Chih-Yueh Chou[2], Tak-Wai Chan[3]
Department of Computer Science & Information Engineering, National Central University, Taiwan[1],
Department of Computer Science and Engineering, Yuan Ze University, Taiwan [2],
Center for Science and Technology of Learning[3]
hon@lst.ncu.edu.tw, ycdeng@lst.ncu.edu.tw, cychou@saturn.yzu.edu.tw, chan@cl.ncu.edu.tw

Abstract. This paper reports a pilot study of how to utilize simulated animal companions to encourage students to pay more effort in their study in the classroom environment. A class of students is divided into several teams. Every student keeps her own *individual animal companion*, called My-Pet, which keeps a simple performance record of its master for self-reflection. Also, every team has a *team animal companion,* called Our-Pet, kept by all teammates. Our-Pet has a collective performance record formed by all team members' performance records. The design of Our-Pet intends to help a team set a team goal through a competitive game among Our-Pets, and promotes positive and helpful interactions among teammates. A preliminary experiment is conducted in a fifth-grade class with 31 students in an elementary school, and the experimental results show that there are both cognitive and affective gains.

Keywords: learning companion, open student model, motivation

1. Implications of Tamagotchi phenomenon in learning

"Motivation is relevant to learning, because learning is an active process requiring conscious and deliberate activity. Even the most able students will not learn if they do not pay attention and exert some effort" (Stipek, 2001). Motivation significantly influences learning, and how to stimulate learners to pay more effort in their learning activities is an important issue. However, pet keeping is a pervasive culture across gender and nationality over a long period, and some studies have observed that pet keeping is naturally attractive to children. The relationships built between pets and their owners are easily elicited based on the human's attachment to pets (Beck & Katcher, 1996; Levinson, 1969). Children clearly have a special bond with their pets, and some researchers believe that children are naturally attracted to pets because they all share the same personality, such as cute, simple and straightforward behaviors (Melson, 2001). With the attachment to pet, children not only feel the feeling of be-loved, be-needed, and other emotional support from pets, but they also tend to respond their love, and taking care of them. Other works also note that interaction with animals increases the social competence and learning opportunities of children (Beck & Katcher, 1996; Myers, 1998). With technology advancement, some technological substitutes for pets have been created. One example is the well-known Tamagotchi (Webster, 1998; Pesce, 2000). Although it is merely simple animated pictures and some buttons, children are quite devoted to the process of nurturing a virtual chicken, caring for it from an egg to a mature rooster.

Our work was inspired by the idea of applying Tamagotchi from pure entertainment to educational field as well as the work on learning companion, a simulated agent that mimics the student herself and provides companionship to the student (Chan, 1996). Animal companions are one kind of learning companions especially designed for pupils. What are implications of Tamagotchi phenomenon in learning? There is an array of interesting research questions to answer on how animal companion may affect self-beliefs, cognitive domain, affective domain, and social domain on students as well as system design. There are two different perspectives towards how people develop beliefs about themselves on motivation and achievement: intelligence or effort (Dweck, 2000). Does more effort significantly affect more positively on learning performance? Is it important for students in the process of developing the belief that success depends on effort rather than on intelligence if they experience that paying effort really contribute to better learning performance? If the answer is yes, then when we deploy animal companion in learning, can we reinforce that belief and become an attitude of students? it further involves several research questions: (1) How to design a goal and a set of mechanisms when incorporating animal companion into a learning environment that will motivate learners to engage in the learning activities and promotes more learning effort for achieving their goal? (2) Besides individual effort, could the learning environment also promote group learning efforts (e.g. mutual monitoring and encouragement for individual learning as well as collaborative learning? (3) For a student, besides feeling responsible for taking care of her animal companion, a healthily growing animal companion also represents her pride and achievement. How these affects impact her learning and can these be under her control? (4) Could such technological substitutes for pets also have the same benefits described above as the real pets? In this paper, the former two questions are focused, and the other two questions will be addressed in our following investigation.

For answering these research questions, a simple version of My-Pet system (Chen et al., 2001; 2002; Chen et al., 2003) is developed and deployed to EduCities (Chan et al., 2001; Chang et al., 2003), a web-based environment in the internet, for testing its initial feasibility. In this study, we have the improved version of My-Pet integrated with Our-Pet system to form the My-Pet-Our-Pet system. The system was used in an elementary classroom. We believe that My-Pet-Our-Pet is a rather specific example of adopting the concept of open student model (Self, 1988; Kay, 1997; Bull, 1998; Bull, 2004), a means for extending application range of student modeling. The learners are grouped into several teams, in which each learner is surrounded by two kinds of animal companions: *individual animal companion*, called My-Pet, and *team animal companion*, called Our-Pet. The repertoire of activities in My-Pet-Our-Pet consists of four modes: nurturing My-Pet mode, learning mode, inspecting My-Pet-Our-Pet mode, and team Our-Pet competition mode.

2. My-Pet-Our-Pet

2.1 My-Pet

Nurturing My-Pet mode: My-Pet is a computer-simulated pet needing a student's nurture and care. In order to take good care of My-Pet, the student needs to make effort to learn so that she can earn the pet's food and eligibility to use some caring tools. For example, while My-Pet's energy level is low because it is hungry, the student has to spend her "coins" to buy food. However, these "coins" are designed to be earned according to the amount of effort paid by the student in the learning activity. In this mode, My-Pet plays two roles: *motivator* and *sustainer*. Based on the student's attachment to My-Pet and good will for it, the student is motivated to take action to learn. The good will is the cause and learning is the effect. Such design is similar to what Rieber called "sugar coating" (Rieber, 1996). Although this initial motivation for learning is not for the purpose of learning itself, however, if the student later finds that the subject matter required for learning is an intriguing and rewarding experience, this initial motivation may change qualitatively to motivation for learning this subject matter itself. In addition, pet keeping is a regular and long-term activity. With appropriate reinforcement, My-Pet may be able to sustain some desired student behaviors to become a habit. It is quite possible that nurturing My-Pet is the real intention of the students and learning is just happened to be a side-effect in the process of nurturing. This mode is sort of "package" mode for subsequent learning activities.

Learning mode: The learning task is to learn about and apply idiomatic phrases. A student could read the historical story to understand the original meaning, identify the key words and key sentences, and then practice the application of these phrases in different contexts. An important component of My-Pet is its master's performance record. It is recorded in two levels: *domain* and *attribute*. Domains include cognitive, emotional, and social domains, as shown in Figure 1. For cognitive domain, My-Pet adopts a simple overlapping modeling approach, and there are three attributes: "remembering", "understanding", and "applying," with values are numerically recorded according to student's mastery level. Furthermore, the representation of attribute values of cognitive domain has two levels: detailed value and summarized value. The detailed value is presented aside each phrase, and the summarized value is the aggregation of the detailed values. This information makes the student quickly aware of her own performance about the learning task in the activity.

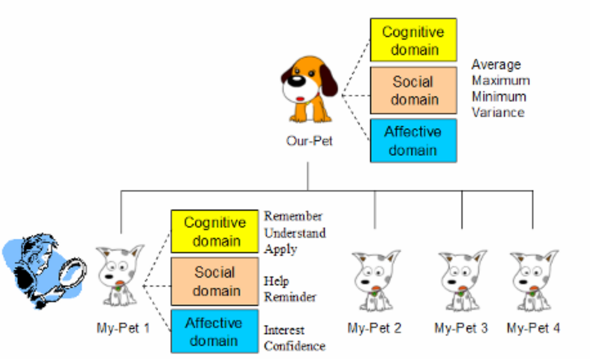


Figure 1. Inspecting performance records reflected by My-Pet and Our-Pet

The emotional domain consists of two attributes: "confidence" and "interest". "Confidence" is evaluated by the rate of successes of answering questions correctly, and "interest" is determined by the frequency the student involved in learning activities of a topic even if she is not asked to do so or after class. With this information, the student could grasp easily the sense of how much effort she has paid. In the social domain, there are two attributes "reminding" and "helping" recorded according to student's interactions among teammates. The attribute values are collected by an honor system in current version, that is, the student reports to My-Pet how many times she "reminds" or "helps" her teammates to study in each session. Moreover, for helping students understand their situation with impression, My-Pet's emotional status and passively-initiated dialogues are designed to disclose the status of three domains based on some heuristics. For example, if a student's value in cognitive domain is low, My-Pet's mood will be sad. If the student initiates a conversation with My-Pet, it will tell the student what is the cause of its sadness. In this mode, My-Pet plays the role of self-*reflector*. Self-reflection through viewing the "internal" representation of My-Pet, which is essentially the performance record of the student in different domains, can help the student look at herself and hence understand herself better or enhance her self-awareness. In other words, My-Pet is sort of the mirror of the student. While the student looks at this performance record of My-Pet, she actually observes the result of her own learning effort.

2.2 Our-Pet

Inspecting My-Pet-Our-Pet mode: Our-Pet is a team's pet that is commonly owned by four teammates. An important component of Our-Pet that largely governs the behavior of Our-Pet is a collective performance record, "inspectable" by all members. There are three levels of the collective performance record: domain, attribute, and viewpoint. The domains and attributes are the same as those in My-Pets. For each domain and each attribute, there are four kinds of viewpoints: "average", "minimum", "maximum", and "variance." Through "average" viewpoint, a student may view the average status of her team's mastery values in the cognitive domain so that she can know the team's overall situation. Through "minimum" viewpoint, all teammates can view mastery value of the weakest teammate, and other teammates will then naturally be urged to "help" or "remind" the weakest one to do more remedial work. Through "maximum" viewpoint, the strongest teammate's value will be observed, and it encourages the strongest one to do more for enrichment and strives for excellence, but this will increase their "variance." Therefore, it also urges the stronger teammates to help the weaker teammates so that they can narrow their gaps. The mechanisms for affective and social domains are similar to that of the cognitive domain. To provide different perspectives to promote self-reflection, Our-Pet's passively-initiated dialogues are designed to express the different statuses between My-Pet and Our-Pet in three domains based on the rule-based mechanism. For example, if a student finds her My-Pet's values in the cognitive domain are low. She may talk to Our-Pet, which then prompts the student what situation her performance is, what situation the team performance is, and what actions she can take to improve.

In this mode, My-Pet and Our Pet plays two roles: self-*reflector* and improvement *indicator*. Different from the reflector role played in the learning mode in which a student can only inspect her My-Pet, the student in this mode could observe both My-Pet and Our-Pet, and self-reflection is consequently further promoted. Moreover, by comparing these different perspectives of information, she knows what she has mastered, what she has not mastered, what other teammates have mastered, what other teammates have not mastered, and the directions to improve her current status or help other teammates.

Our-Pet competition mode: Our-Pets involve in a series of team competition games. Winning or losing a game will depend on attribute values of the two competing Our-Pets. Each game has four rounds of contests. The final result of a game is calculated by accumulating the results of four rounds and there is a ranking of all teams. A student represents her team in one round will rotate three turntables to determine which domain, which attribute, and which viewpoint of Our-Pet to compete against the other team. In other words, the chance of Our-Pet winning the game depends on some attribute values of teammates. To increase winning chance, it demands the whole team's effort to improve all these attribute values. Team competition of Our-Pets forms the situation of intra-team collaboration, helps the whole team establish their common goal, and urges all teammates to work hard for learning. Moreover, it promotes the collaboration which not only needs individual accountability in the team, but also encourages positive and helpful interactions among the teammates. Therefore, in this mode, the roles of Our-Pet are *goal setter* and *motivator* for promoting both individual and collaborative effort for learning.

3. Experiment

A preliminary experiment of My-Pet-Our-Pet was conducted in a classroom of Wang-Fang elementary school (see Figure 2) from November 2004 to January 2005. The classroom was a one-on-one classroom environment, that is, every student in this classroom has a computing device with wireless capability (see www.g1on1.org). Due to the constraint of regular school timetable in Wang-Fang elementary school, comparison of influences of My-Pet and Our-Pet on students still need further assessment. The objective of this experiment mainly focuses on evaluating the learning effect and affective influences of My-Pet-Our-Pet. The subjects were 31 fifth-grade students and they were arranged to eight 4-children teams (except the eighth team only has 3 students) with their academic performance well-mixed, that is, each team had one high-performance student, two mid-performance students, and one low-performance student. The experiment was divided into two phases, and each phase students used Tablet PCs for 10 fifteen-minute sessions in the class for one and a half months. However, only learning material was provided in the first phase for the control group, and both learning material and My-Pet-Our-Pet were provided in the second phase.

Figure 2. Snapshots of My-Pet-Our-Pet used in the 1:1 classroom environment

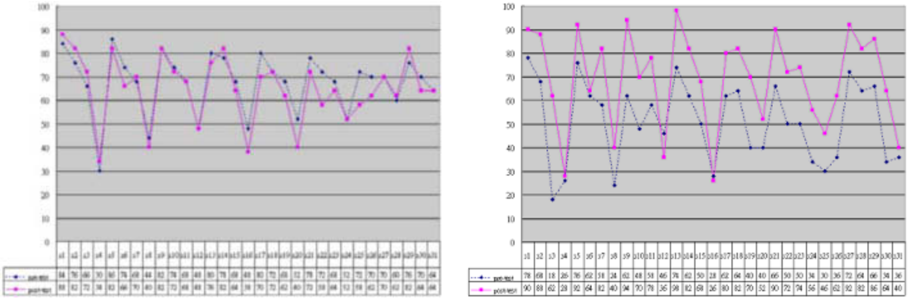
We addressed two questions, one in cognitive domain and one in affective domain, in this experiment. The cognitive question is: what are the learning effects after students use My-Pet-Our-Pet? The affective question is: what about their affective experience of using My-Pet-Our-Pet in the classroom environment? For the cognitive question, pre-test and post-test were administered for forty minutes in each phase. Each test has fifty items and contains three categories of questions: memorizing, understanding, and applying. To collect affective experience data, face-to-face interviews in the classroom were taken for further analysis and discussion.

3.1 Results

The results of pre-test and post-test in the two phases are illustrated in Figure 3. Figure 3 (a) is the score distribution of the first phase, where the pre-test (blue dotted-line) and post-test (red concrete-line) are almost the same. However, in figure 3(b), score distribution of the second phase was obviously different, where most of the scores in post-test were higher than pre-test, and is statistically significant (p<.005) in the paired-sample test, as shown in Table 1.

Table 1. Result of paired sample test in two phases

	Mean	Std. Dev.	T	p(Sig.)
Post-test – pre-test in 1st phase	-2.65	5.782	-2.547	.016
Post-test – pre-test in 2nd phase	18.19	10.663	9.5	.0001



(a) scores of pre-test and post-test in 1st phase (b) scores of pre-test and post-test in 2nd phase

Figure 3. Scores of pre-test and post-test in two phases

3.2 Feedbacks from interview

For collecting students' affective experience, interviews are conducted and summarized as following. First, in the mode of "nurturing My-Pet", while students were asked to compare their feelings in the two phases, 31 students were all impressed and enjoyed in the process of raising My-Pets.

"I like pets very much, but I can't keep pets in at home. In My-Pet-Our-Pet, I can take care of my own pets, and it is very interesting." (student #34)

"It (My-Pet-Our-Pet) uses many ways to make us feel that learning idiomatic phrases is an appealing task." (student #28)

"My learning progress has doubly increased. I love My-Pet-Our-Pet because I not only can learn idiomatic phrase, but nurture pets." (student #27)

Second, in the learning mode, when the students were asked to compare their engagement in these two phases, 2 students expressed that they were all the same to them, because they felt that learning idiomatic phrases is boring. 29 students stated that they were more engaged in the reading session in second phase.

"I will take it seriously, because I want to earn coins to nurture my pet." (student #22)

"Of course, I must pass the assessment, and then I could gain the coins." (student #12)

Besides, 26 students felt that My-Pet's emotional expression is an effective way to convey information and learning status to its master, and further affects the students' behavior, especially taking initiative to learn.

"When I'm seeing My-Pet's mood is happy, I feel better too. But when it was depressed or unhappy, I would think what's wrong? Then taking it along to buy candies with coins, to learn idiomatic phrase, and it will be happy." (student #27)

"If my pet is sad, I will also feel unhappy. It seems to be my real pet" (student #13)

Third, in the inspecting My-Pet and Our-Pet mode, we asked "whether the inspecting functions provided by My-Pet and Our-Pet are helpful to you?" 27 students feel that they are convenient ways to understand their own learning statuses.

"I care its (My-Pet's) status, because its status is equal to my learning status." (student #21)

"I frequently see the average values of Our-Pet, and it lets me know what our team's situation is. Then I go back to study hard for earning coins." (student #25)

"When seeing my value is the highest among four people, I encourage them. I had encouraged all our teammates." (student #27)

Finally, in the Our-Pet competition mode, the question is: "how team competition of Our-Pet affects the interaction with other teammates?" 4 students rarely care about team competition; 27 students are affected by Our-Pet competition. (15 students felt that team competition was the matters of honor and solidarity, and hence facilitated their communication and interaction. However, other 12 students seldom interacted with other teammates, but learned harder individually.)

"In the beginning, our team's competitive ranking is the last, and then becomes the fifth. Because of that, I tell them (other two boys) to study more for raising the values, to earn coins harder." (student #2)

"We (students #33 & #22) discussed the idiomatic phrase together. Sometimes we two girls answered the question together, and sometimes one found out the answer, and the other responded." (student #33)

3.3 Discussion

According to the results of experiment, we found that all 31 students were engaged and enjoyed in raising My-Pet, and 29 students were willing to pay more learning efforts to improve their learning progress reflected by My-Pet and Our-Pet. Consequently, they earned better academic performances. Moreover, in order to win in the team competition of Our-Pet, 15 students were often monitored and encouraged each other while learning. In other words, the design of My-Pet-Our-Pet had promoted the individual's learning effort, and group learning effort. However, regarding to collaborative learning among teammates, it seldom happened. What were the reasons? Analyzing the content of students' dialogues, we found that topics of "what should we name our team?" or "which team should we select as our opponent?" were more popular. For team competition, against our expectation, most students went back to study harder by themselves, rather than having more interactions (collaboration) with other teammates.

There are some possible reasons: (1) Learning activities that need all members' decision could trigger discussion and collaboration, and the four modes in My-Pet-Our-Pet lack such designs. (2) If the roles played by teammates were more diversified and each role is essential for winning, then it facilitated more teamwork. In My-Pet-Our-Pet, the teammate's roles were the same. (3) There are no findings to support the original hypothesis: the stronger tends to help the weaker for team competition. Team's ranking indeed stands for teammates' honor, but some factors also have significant influences, such as students' personality (if a student is shy and introvert, then she may not be very social), gender difference (girls like to play with girls, rather than boys), and friendship (some students ask us why couldn't let them find their good friends to form a team).

4. Conclusion

In this paper, we described and discussed the design rationales of a system called My-Pet-Our-Pet which does not only encourages students to work hard in learning, but also promotes helpful interactions through the representation of the individual and the collective performance records kept in My-Pet and Our-Pet, respectively. The preliminary results show that all 31 students indeed were engaged and enjoyed in the process of raising their pets, and most of them (29 students) paid more effort to improve their learning statuses reflected by My-Pet and Our-Pet, and academic performance improvement is statistically significant by comparing the two successive phases. Furthermore, teams' learning efforts were also promoted. About half of students (15 students) would mutually monitor and encourage each other to achieve their common goal. The quality and the design of interactions in collaborative learning should be enhanced and enriched because compared to the pure

Web-based virtual environment, learning in the classroom environment, where the personal interactions are direct, is more complex. To address these issues, more formal evaluations are required.

Most people conceive computer as a tool. Artificial intelligence researchers intend to make computer more than a tool. A candidate for them to pursue this goal is intelligent agent, which is required to be autonomous so that it can take initiative to interact with its user. On the contrary, for animal companion, a student takes a much stronger initiative for interacting with it. This is because users have a model on any entity they are interacting with. The animal companion is portrayed as a pet in real lives, urging a student's innate drive to nurture it. Animal companion is not an autonomous agent, though in some occasions it can or should, nor a tool. Even there is a role of tool in animal companion, it is implicit and is used, at least on the surface, only for the sake of taking care of the animal companion itself.

Learning achievement is usually what a student cares about most, through which her self-concept and identity develop. Now, her animal companion is another thing the student cares about, so much as if it were her second identity. Furthermore, animal companions serve as "mirrors" on which a student interacts with in meaningful and fruitful ways, supporting active self-reflection on cognitive, affective and social domains.

References

[1] Beck, A., and Katcher, A. (1996). Between pets and people. West Lafayette, IN: Purdue University Press.

[2] Bull, S. (1998). 'Do It Yourself' Student Models for Collaborative Student Modelling and Peer Interaction, in B.P. Goettl, H.M. Halff, C.L. Redfield & V.J. Shute (eds), *Proceedings of International Conference on Intelligent Tutoring Systems*, Springer-Verlag, Berlin Heidelberg, 176-185.

[3] Bull, S. (2004). Supporting Learning with Open Learner Models, *Proceedings of 4th Hellenic Conference with International Participation: Information and Communication Technologies in Education*, Athens, Greece. Keynote.

[4] Chan, T.W. (1996). Learning Companion Systems, Social Learning Systems, and the Global Social Learning Club. *International Journal of Artificial Intelligence in Education, 7(2)*, 125-159.

[5] Chan, T.W., Hue, C.W., Chou, C.Y., & Tzeng, O.J.L. (2001). Four spaces of network learning models. *Computers and Education, 37,* 141-161.

[6] Chang, L. J., Yang, J. C., Deng, Y. C., Chan, T. W. (2003) EduXs: Multilayer educational services platforms. *Computers and Education 41(1)*, 1-18.

[7] Chen, Z. H., Deng, Y. C., Chang, L. J., & Chan, T. W. (2001). An motivating learning platform for children through the pet-raising mechanism. National Computer Symposium, Taipei, 203-210.

[8] Chen, Z. H., Deng, Y. C., Chang, L. J., & Chan, T. W. (2002). An approach to cultivating reading habits for children with pet-raising games. *The 6th Global Chinese Conference on Computers in Education*, Beijing, 213-216.

[9] Chen, Z. H., Yang, J. C., Deng, Y. C., & Chan, T. W. (2003). Environment design through coupling pet-raising games with domain-independent learning activities. *The 7th Global Chinese Conference on Computers in Education*, Nanjing, 755-759.

[10] Dweck, C. S. (1999). Self-theories: their role in motivation, personality, and development. Philadelphia: Taylor & Francis.

[11] Melson, G. F. (2001). Why the wild things are: Animals in the lives of children. Cambridge, MA: Harvard University Press.

[12] Pesce, M. (2000). The playful world: how technology is transforming our imagination. New York: Random House.

[13] Stipek, D. J. (2001). Motivation to learn: integrating theory and practice (4th ed.). Boston: Pearson Allyn & Bacon.

[14] Webster, N. C. (1998), Tamagotchi, *Advertising Age, 69(26)*, 43.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

ArithmeticDesk: Computer Embedded Manipulatives for Learning Arithmetic

Hercy N.H. Cheng*, Ben Chang**, Yi-Chan Deng*, Tak-Wai Chan***
**Department of Computer Science and Information Engineering,*
***Research Center for Science and Technology of Learning,*
****Graduate School of Network Learning Technology*
National Central University, Taiwan

Abstract. Physical manipulatives have been applied in traditional education for a long time. This paper proposes that by embedding computing power in manipulatives computers can monitor students' physical manipulations to support learning. This paper also describes the design of a digital desk prototype, called ArithmeticDesk to illustrate the vision of computer embedded manipulatives and takes learning fractions as an example. The study is an attempt to accommodate physical and virtual manipulations, as well as to eliminate the gap between traditional and computer-based learning activities. More experiments and studies will be conducted in the future.

1. Introduction

Manipulatives, which are small tangible objects that students can manipulate by hands, have been extensively used from kindergarten to middle schools. If well-designed, physical manipulation of manipulatives can improve students' conceptual understanding. Especially when learning mathematics, students can build their abstract knowledge by the aids of manipulatives. In practices, blocks, beads, ropes, sticks, and so forth are conventional learning manipulatives. Generally speaking, blocks and beads can enhance the sense of number while ropes and sticks can support the measurement of length. Besides, some manipulatives with physical constraints can scaffold learning. For example, in Asia, abacuses are not only traditional calculators but also manipultaives for learning integers and the decimal system. Ullmer and Ishii [15] described the beads and the constraints (i.e. the rods and the frame) of the abacus as "manipulable physical representations of abstract numerical values and operations." In other words, students can touch, feel and manipulate the digits physically. Unfortunately, in places populated with Chinese, the use of abacuses in class is gradually disappearing.

As computers become common learning devices in classrooms, there is a gap between physical and virtual environments. Our research field seems to have long been governed by treating the bare computer either as a learning tool, a mediator for communications between 'person-to-person' for supporting social learning, or an intelligent learning environment. As the era of ubiquitous computing is approaching in which simultaneous communications between 'person-to-many-everyday-objects-around' enabled by technology of wireless sensor networks become commonplace, our research on embedded learning manipulatives may help shed light to a new research avenue, in addition to reaffirming the contribution of traditional physical learning manipulatives to mathematical education.

2. Related Works

This study attempts to link two different researches. The first is to survey how the manipulatives were applied in mathematical education, and the second is to explore the technologies of computer embedded manipulatives.

2.1 Manipulatives in Learning Mathematics

In Post's survey [11, 12], Lesh (1979) argued that manipulative materials can be regarded as an intermediary between the real world and the mathematical world. In other words, students can use manipulatives to learn mathematical concepts from concrete but complex situation toward simplified but abstract symbolic operations. On the other hand, Kolb [5] proposed the "experiential learning theory" to stress the role of concrete experiences in the learning process. He also argued that the experiential learning cycle – concrete experience, reflective observation, abstract conceptualization, and then active experimentation – can explain how people learn from experiences and apply them to new situation.

Martin and Schwartz [8] identified "physically distributed learning," one of the ways of physical manipulation to support thinking and learning. They examined how the process can support children's development of fraction concepts, and found that manipulating physical materials helps children adapt the environment and facilitate their reinterpretations. They also found that children can develop such abilities in adaptable environments (such as tiles) better than in well-structured environments (such as pies).

Referring to Bruner's model of mathematical ideas [2], Lesh also proposed that five modes of representation – real world situations, manipulative aids, pictures, spoken symbols, written symbols – should be considered and adopted interactively [12]. The Rational Number Project [16], supported by American National Science Foundation, has utilized and corroborated Lesh's model on teaching rational number concepts. Therefore, when looking into manipulatives, we should support the interdependence between manipulatives and the other four representations. For example, while fractional symbols, pie charts (pictures), or applied examples (real world situations) are presented, students should have the ability to show the same values of fractions by operating blocks (manipulatives). On the other hand, students should also be able to interpret manipulatives into other modes of representations, including other forms of manipulatives.

2.2 Computer Embedded Manipulatives

In the field of human computer interactions, the researches on tangible user interfaces (TUIs) [4] has been drawn more and more attention. TUIs are interfaces of physical objects and environment coupled with digital information, taking advantage of humans' inherent tactile sense, as compared with graphical user interfaces (GUIs).

Our work was inspired by Sensetable [10] developed in MIT media lab, which is a sensory table allowing users to control graphical figures by manipulating tangible objects tracked electromagnetically on a tabletop display surface. Tangible Viewpoints [9], one of the applications of Sensetable, is designed to navigate and explore stories by handling different characters in the stories. The PitA Board [3] developed in the Center for Lifelong Learning and Design (L^{3}D) at University of Colorado which allows inhabitants to participate in design of their town in face-to-face setting by manipulating small models of buildings and by drawing the roads and bus routes. For implementing the sensory screen, the PitA Board first adopted SmartBoard technology and then electronic chessboard technology. Sugimoto, et al. developed ePro [14], a system with a sensor-embedded board, to support face-to-face learning. ePro using RFID technology allows students to simulate the environment by manipulating objects of trees, houses and factories. Another application of the sensing board is Symphony-Q [7], which enhances children's music experiences through creating chords and music collaboratively by hands. These related systems reveal that the area of computer embedded manipulatives is coming to maturity, and it is time to apply such technologies to formal and informal educations.

3. Computer Embedded Manipulatives for Learning Fractions

This study chooses learning fractions to investigate the capacity of computer embedded manipulatives for learning arithmetic. Previous researches of computer-based environments supporting learning fractions revealed that one of the advantages of computers is to show the graphical representations of fractions and to exhibit the operation of partitioning [1, 6, 13]. Nevertheless, when considering computer embedded manipulatives, we can use both strengths of GUIs and TUIs to counterbalance the weaknesses of each other [3]. For example, the perceptibility and manipulability of physical materials contribute to ample tactile interactions, while the visualization, simulation and animation of virtual materials give students clear and immediate interpretation. Table 1 shows several complementary strengths of TUIs and GUIs in learning fractions. Generally speaking, by manipulations learners are engaged in concrete experiences, and by visualization learners can build abscract conceptualization. In other words, without either one, it is potentially difficult for learners to make the transition from concrete experiences to abscract conceptualization.

Table 1. The strengths and weaknesses of tangible and graphical objects in fraction learning

Factors	Tangible Objects	Graphical Objects
To perceive the dimension of objects	The volume and weight	Only the plane
To move objects	By hands	Only by mouse
To partition objects	Only if divisible	No limits
To unitize objects	Unclear (Placing objects in proximity)	Clear
To get feedbacks	Without immediateness	By immediate feedbacks
To build the concepts	By realizing the real situations	By making relations to symbols
To apply the concepts	By active manipulations	By simulating the results
To collaborate with mates	Good in face-to-face settings (Intermediaries of communication)	Weak

In fraction learning, partitioning and unitizing manipulatives are two major physical actions [8]. However, physical objects fall short of partitioning because it is difficult to design a solid physical object which can be partitioned arbitrarily like graphical representations. A substitute action can be designed to make several totally equal but fairly partitioned physical objects to replace the original one.

In some cases, the operation of fractions involves unitizing – treating objects in each partition as a unit. For example, $8 \times \frac{3}{4}$ represents partitioning eight objects equally into four parts and then taking three parts. In this case, each part has two objects, and students have to regard every two objects as a unit so that they can understand the meaning of "three". Students perform such unitizing mentally, and the manipulatives lack explicit physical constraints to support this action. Graphical user interfaces can complement the deficiency – drawing a circle under objects in each partition, for example.

Physical manipulatives have potential to help students analogize from real situations and interpret them, whereas dynamic graphical simulation improves students' comprehension of mathematical symbols. In order to overcome the gap between physical manipulatives and graphical representations, we propose that computers ought to “perceive” what students do with manipulatives, or even “understand” why they do that.

4. System Design

From Lesh’s perspective, ideally we should integrate the five representations to support fraction learning. However, to perform a simple version of our system, this study is only centered on manipulatives, pictures and written symbols.

4.1. Hardware architecture

Figure 1 shows the hardware architecture of ArithmeticDesk or ArithDesk for short. The tabletop is a large sensor board for detecting the manipulatives by electromagnetic technologies. Every manipulative has a micro switch at the bottom, so that when a student takes up or puts down a manipulative, the circuit transitorily emits electromagnetic signals of its identification to the sensor (Figure 2) and then is immediately switched off to avoid interfering with each other. The sensor board transfers the identification with the position of the manipulative to the embedded computer. After processing the data, the output is displayed on the tabletop through the projector.

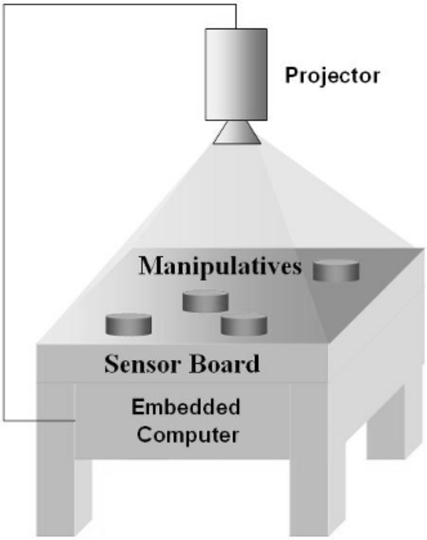


Figure 1. Hardware architecture of ArithmeticDesk

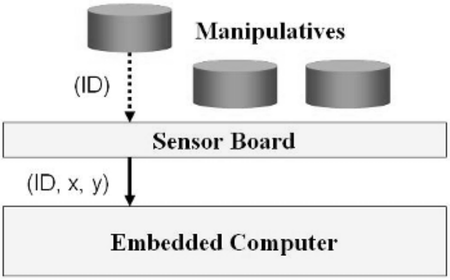


Figure 2. Input mechanism of ArithmeticDesk

4.2. Software architecture

The software architecture of ArithDesk is shown in Figure 3. **Manipulatives control module** collects the identification and position for each manipulative, and records the position in a *position table* according to the identification. The *position table* is referred to another *definition table*, in which the fractional meaning, graphical representation, and its physical dimension of every manipulative are recorded. At the same time, all manipulations are also logged in database for further analysis.

When students finish the arrangement of manipulatives, **fraction knowledge module** retrieves the identification of those manipulatives in the range of tablet from the *position table*, finds their fractional meanings from the *definition table*, and then checks if the arrangement is correct. On the other hand, **Pattern analysis module** analyzes the logs from the database to generate some patterns of mis-manipulations. These patterns help the system to find the misconceptions that the students could have.

According to the correctness of current manipulation and the patterns of mis-manipulations, **instruction module** provides appropriate instructions. For example, if a student wants to partition 7 tiles equally into 2 piles, but he put 3 tiles in one pile and 4 tiles in the other, then the system provide a hint that such partition is not fair.

Visualization module receives the results of the instructions, and updates the instructions on the screen. Additionally, this module also retrieves the positions of every manipulatives in the range of tablet from the *position table*, and updates their corresponding graphical representations dynamically.

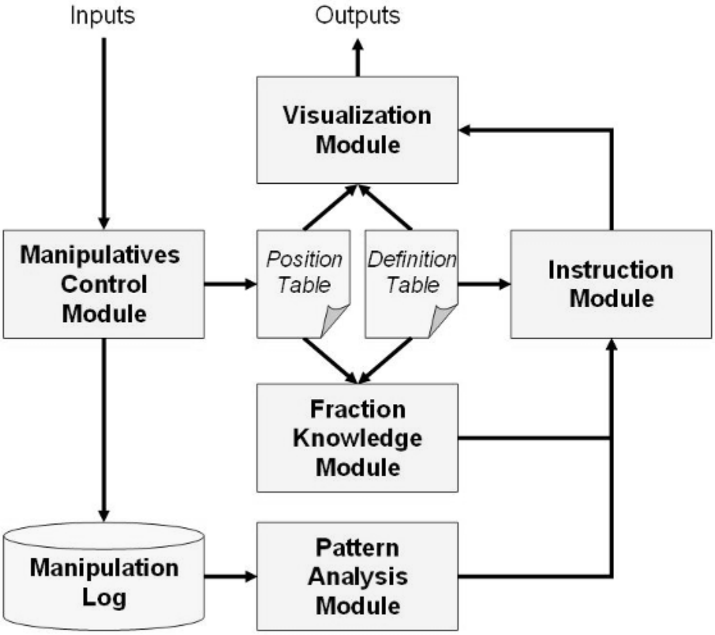


Figure 3. Software architecture of ArithmeticDesk

4.3. Supports for fraction learning

Figure 4-7 depict how the system supports learning fraction concepts and operations. All cuboids and cylinders in these figures represent manipulatives.

Naming fractions.

Figure 4 shows how the system scaffolds students to name a fraction. On the screen, the system displays *partition grids*, surrounded by a *whole frame* (Figure 4(a)). The *partition grids* help learners unitize manipulatives within each grid as a partition, while the *whole frame* gives them another visual support that a whole consists of several partitions.

At first, the system instructs the learner to identify the whole by gathering manipulatives in a *whole frame* (Figure 4(b)). Then the system displays *partition grids* which have equal number of grids to the denominator to help the learner partition the whole. When he puts one or more manipulatives in a grid, the system further displays the same *imprints* in all grids according to the shape and number of the manipulatives. The *imprints* guide the learner to think that every grid should have equal shape and number of manipulatives. If putting different shapes or numbers of manipulatives in grids, he possibly has a misconception of equally partitioning. In addition, as soon as the learner put manipulatives the same as the *imprints* in the grid, the system will color the grid to show that the manipulation satisfies the unit fraction. At the same time, the system also displays the fraction symbol beside the manipulation area to help the learner connect the manipulation to the symbol – the denominator is the number of all grids, and the numerator is the number of colored grids.

The *partition grids* and the *whole frame* imply that a fraction is not an absolute number but relative to the whole. In other words, no matter what the manipulatives are and no matter how many manipulatives the whole represents, a fraction can be presented through such manipulations with the visualization (Figure 4(c)). Therefore, the system has the capacity for providing various forms of manipulatives, such as cuboids, cylinders, beads, or other artifacts.

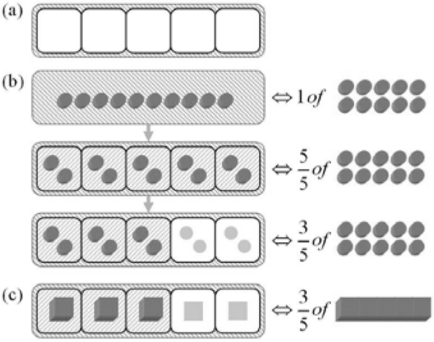


Figure 4. Naming fractions.
(a) Partition grids with a whole frame.
(b) The process of manipulation for naming fraction
(d) An example using different representations.

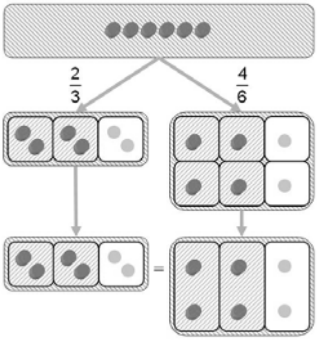


Figure 5. Renaming fractions.

Renaming fractions.

Renaming fractions is an important skill in the learning of fraction operations. Figure 5 shows an example of renaming a fraction. At first, the system instruct the learner to use manipulatives to create two fractions which are equivalent for the same whole but have different denominators – $\frac{2}{3}$ and $\frac{4}{6}$, for example. After the learner finish the manipulation, the system modifies the *partition grids* into similar forms so that the learner can find both have the same results, that is, they are equivalent. Then the system displays the *partition grids* once again, and instructs him to observe the differences of the two fractions – if the number of grids (denominator) is doubled, then the number of grids with manipulatives (numerator) is also doubled, but the size or number of manipulatives within each grid becomes a half. The learner can test the hypothesis by tripling the denominator. Such observation gives them an interpretation of the changes of the denominator and numerator when renaming a fraction symbol.

Multiplication of fractions

The system considers the multiplication of fractions as two cases: the multiplier is an integer or a fraction. The former represents duplicating the multiplicand, while the latter represents partitioning the multiplicand equally and taking several partitions. In the former case (Figure 6(a)), the system instructs the learner to reproduce the fraction by using manipulatives, and then to add all fractions together, like the multiplication of integers. By contrast, the latter case is more complicated. Figure 6(b) shows an example that the multiplicand is multiplied by a fraction. In this case, because it is difficult to partition $\frac{1}{3}$ directly, the system instructs the learner to use manipulatives to rename the multiplicand as $\frac{2}{6}$ so that the learner can partition these manipulatives into two parts and taking one part.

Division of fractions

Fraction division perhaps is the most difficult concept and operation to learn in fraction learning. The system describes a division equation $A \div B = C$ as the statement "partitioning A equally into several piles and each pile has B, so C piles can be created." Because the division statement is equivalent to the definition of fraction, the system can make use of the *whole frame* to help learners identify the divisor. Figure 7 shows a simple example of division of fractions. Firstly, the system instructs the learner to use manipulatives to construct the dividend and divisor in accordance with the same whole, and then tells the learner the division statement. Following the statement, the system removes the *partition grids* and the *whole frame* of the dividend in order to guide the learner to think of the dividend as the elements to be partitioned. As regards the divisor, the system re-creates a *whole frame* which surrounds the *imprints* so that the learner can identify the new whole. Then the system instructs the learner to partition the manipulatives of the dividend into the new *whole frames*. If the learner can not tell the answer, he can choose to activate the *fraction grids* to help him recognize the number of piles.

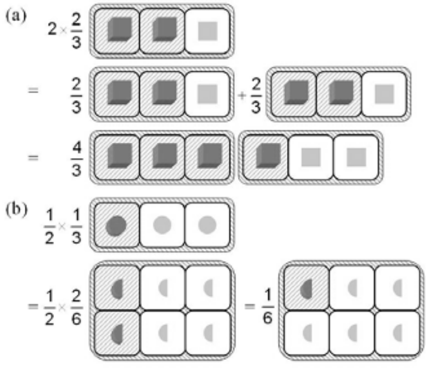

(a)
$2 \times \frac{2}{3}$
$= \frac{2}{3} + \frac{2}{3}$
$= \frac{4}{3}$
(b)
$\frac{1}{2} \times \frac{1}{3}$
$= \frac{1}{2} \times \frac{2}{6}$
$= \frac{1}{6}$

Figure 6. Multiplication of fractions.
(a) An integer × a fraction
(b) A fraction × a fraction

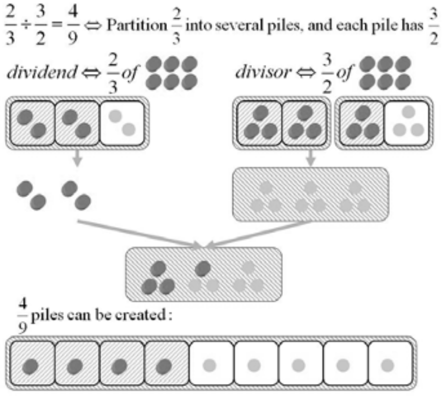

$\frac{2}{3} \div \frac{3}{2} = \frac{4}{9} \Leftrightarrow$ Partition $\frac{2}{3}$ into several piles, and each pile has $\frac{3}{2}$
dividend $\Leftrightarrow \frac{2}{3}$ of
divisor $\Leftrightarrow \frac{3}{2}$ of
$\frac{4}{9}$ piles can be created:

Figure 7. Division of fractions.

5. Future works and conclusion

This paper presents the prototype of ArithDesk to illustrate our vision that computer embedded manipulatives bridges the mathematical education and computer-based environments. Learning fractions is only a starting point. Currently, we are working on development of the ArithDesk prototype as well as planning to collect the data for supporting the analysis of learners' manipulation. On the other hand, we also expect that computers should be an *augmented technology*, seamlessly integrated into almost every, if not all, objects in a classroom, unobtrusively enhancing daily learning activities, rather than being an independent entity by itself. Moreover, we envision that such technologies could lead teachers and students back to learning with inborn senses. In the near future, continuing the study of computer embedded physical learning manipulatives will lead to the emergence of interesting hidden issues of human computer interactions and their implications in learning science.

References

[1] Akpinar, A., and Hartley, J.R. (1996) Designing interactive learning environments. *Journal of Computer Assisted Learning*, 12, 33-46.

[2] Bruner, J.S. (1966) Toward a theory of instruction, Mass Belknap Press of Harvard University, Cambridge.

[3] Eden, H. (2002) Getting in on the (Inter)Action: Exploring Affordances for Collaborative Learning in a Context of Informed Participation. *Proceedings of the Computer Supported Collaborative Learning Conference CSCL'2002*, Boulder, CO, pp. 399-407.

[4] Ishii, H., and Ullmer, B. (1997) Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. *Proceedings of Conference on Human Factors in Computing systems CHI '97*, (Atlanta, Georgia, USA, March 1997), ACM Press, 234-241.

[5] Kolb, D.A. (1984). *Experiential learning: Experience as the source of learning and development.* Englewood Cliffs, N.J.: Prentice Hall.

[6] Kong, S.C., and Kwok, L.F. (2002) A graphical partitioning model for learning common fraction: designing affordances on a web-supported learning. *Computers & Education*, 40(2), pp. 137-155.

[7] Kusunoki, F., Sugimoto, M., Kashiwabara, N., Mizonobe, Y., Yamanoto, N., Yamaoku, H., and Hashizume, H. (2002) Symphony-Q: A Support System for Learning Music through Collaboration. *Proceedings of the Computer Supported Collaborative Learning Conference CSCL'2002*, Boulder, CO, pp.491-492.

[8] Martin, T., and Schwartz, D.L., Physically distributed learning: adapting and reinterpreting physical environments in the development of fraction concepts, *Cognitive Science* (in press).

[9] Mazalek, A., Davenport, G., and Ishii, H. (2002) Tangible Viewpoints: Physical Navigation through Interactive Stories, *Proceedings of the Participatory Design Conference (PDC '02)*, (Malmo, Sweden, June 23-25, 2002), CPSR, pp.401-405

[10] Patten, J., Ishii, H., Hines, J., and Pangaro, G. (2001) Sensetable: A Wireless Object Tracking Platform for Tangible User Interfaces, *Proceedings of Conference on Human Factors in Computing Systems CHI'01* (Seattle, Washington, USA, March 31-April 5, 2001), ACM Press, pp. 253-256

[11] Post, T. (1981) The Role of Manipulative Materials in the Learning of Mathematical Concepts, In Lindquist, M.M. (Ed.), *Selected Issues in Mathematics Education*, Berkeley, CA:National Society for the Study of Education and National Council of Teachers of Mathematics, McCutchan Publishing Corporation, pp. 109-131.

[12] Post, T., and Cramer, K. (1989) Knowledge, Representation and Quantitative Thinking. In M. Reynolds (Ed.), *Knowledge Base for the Beginning Teacher-Special Publication of the AACTE*, Pergamon, Oxford, 1989, pp. 221-231.

[13] Steffe, L.P., and Olive, J. (1996) Symbolizing as a constructive activity in a computer microworld. *Journal of Educational Computing Research*, 14(2), pp. 113-138.

[14] Sugimoto, M., Kusunoki, F., and Hashizume, H. (2000) Supporting Face-to-Face Group Activities with a Sensor-Embedded Board, *Proceedings of ACM CSCW2000 Workshop on Shared Environments to Support Face-to-Face Collaboration*, Philadelphia, PA, pp.46-49.

[15] Ullmer, B., and Ishii, H. (2001) Emerging Frameworks for Tangible User Interfaces, In Carnoll, J.M. (Ed.), *Human Computer Interaction in the New Millennium*, Addison-Wesley, US, pp. 579-601.

[16] The Rational Number Project, http://education.umn.edu/rationalnumberproject/

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Adaptive Reward Mechanism for Sustainable Online Learning Community

Ran Cheng and Julita Vassileva
Computer Science Department, University of Saskatchewan,
Saskatoon, SK, S7N 5A9 Canada
rac740@mail.usask.ca , jiv@cs.usask.ca

Abstract. Abundance of user contributions does not necessarily indicate sustainability of an online community. On the contrary, excessive contributions in the systems may result in information overload and user withdrawal. We propose a user- and community- adaptive reward mechanism aiming to regulate the quantity of the contributions and encourage users to moderate the quality of contributions themselves. The mechanism has been applied and evaluated in an online community supporting undergraduate students to share course-related web-resources.

Introduction

The proliferation of online communities may lead people to the conclusion that the development of custom-made communities for particular purpose, for example, to support a class, is straightforward. Unfortunately, this is not the case. Although software providing basic community infrastructure is readily available, it is not enough to ensure that the community will "take off" and become sustained. For example, the multi-agent-based synchronous private discussion component of the I-Help system [1] did not enjoy much usage by students and was abandoned in favor of the more traditional asynchronous public discussion forum [2]. A critical mass of user participation was missing in the private discussion forum since the students did not stay constantly logging in the system.

Comtella[*] [3] is a small-scale peer-to-peer online community developed at the MADMUC lab at University of Saskatchewan for sharing academic papers and class-related web-resources among students. Comtella, just like I-Help, depends on a critical mass of participation both in terms of quantity and quality of contributions. Our previous work [4, 5] addressed the problem of motivating students to bring new resources in the system. To achieve a sustainable critical amount of participation, this paper proposes a new adaptive reward mechanism to encourage users to rate contributions thus ensuring decentralized community moderation. The mechanism adapts to the current needs of the community in terms of the number of contributions and also to the individual trends/preferences in the type of contributions of each individual member.

1. Previous work

The problem of ensuring user participation is very important for all online communities [6]. The "critical mass" hypothesis proposed by Hiltz and Turoff [7] states that a certain number of active users have to be reached for a virtual community to be sustained. Our experience with Comtella confirms this hypothesis. In order to stimulate users to make contributions we looked into Social Psychology, specifically in the theories of discrete emotions and of social comparison. We proposed, implemented and evaluated in a case study [5] a motivational approach based on hierarchical memberships in the community (gold, silver, and bronze), awarded to users depending on the quantity of their contributions to the community. The different memberships implied different privileges and prestige in the community. While the case study of using the system to support an Ethics and IT class showed that the motivational strategy effectively increased the number of user contributions, it also seemed to motivate a small number of users to game the system to achieve higher membership levels. They shared many resources that were of poor quality or unrelated to the topic. This made it hard for users to find good resources in the system, resulting in the decreased level of participation in the last week of the study and disappointment reflected in negative user comments in the post-experiment questionnaire.

Our observations mirror the phenomenon called “information overload” [8], which has arisen in some other online communities. It makes users feel swamped by a mass of unwanted information. Jones and Rafaeli [9] found that the users’ most common response is to end their participation in the community, both as contributors and as consumers. Therefore, to create a self-maintaining community, it is necessary to avoid the information overload by controlling the overall number of contributions in the system, motivating users to contribute high-quality resources and simultaneously inhibiting the contribution of poor-quality resources. Therefore, a mechanism of measuring the quality of user contributions is needed.

It is difficult to measure the value of user contributions accurately since quality measures are mostly subjective. Centralized moderation is feasible only for small and narrowly focused communities, where members have very similar evaluation criteria. Therefore, decentralized mechanisms for quality measurement are necessary. One way of evaluating the quality of resources used in online communities like Slashdot [10] is through explicit user ratings. The mechanism has two merits. Firstly, it distributes the task of evaluating resources among the large user pool, thereby making achievable a job that would otherwise have been overwhelming. Besides, the final ratings of resources are more unbiased since they are computed based on ratings from many users. However, a study of the Slashdot rating mechanism [11] showed that some deserving comments may receive insufficient attention and end up with an unfair score, especially the ones with lower initial rating and those contributed late in the discussion. Therefore the timeliness of making a contribution is important and a motivational mechanism should encourage early contributions. This is especially relevant in a class-supporting system like Comtella, or I-Help, since the discussion topic typically change on a weekly basis according to the class curriculum. When the topic is new, it is important to have more contributions, but later it is important to have more ratings to help users cope with the information overload. The needs of the community change in time. Therefore, a motivational mechanism needs to adapt to the dynamic needs of the community and encourage users to contribute early.

The Slashdot study [11] also showed that comments starting their life at a low initial rating have a lower chance to be viewed and rated and are more likely to end up with unfair score. In Slashdot, the initial rating depends on the “karma” of the user who made the comment. The user’s “karma” is his/her reputation for contributing high-quality comments, measured by the ratings his/her previous comments collected. In this way, good comments made by new users or the users who haven’t contributed highly rated comments so far tend not to receive a deserving attention and to collect sufficient ratings to raise the “karma” level of their contributor. This causes a feedback loop resulting in the Matthew effect [12] or “the rich get richer”. A fair rating mechanism should give all contributions an equal chance at start.

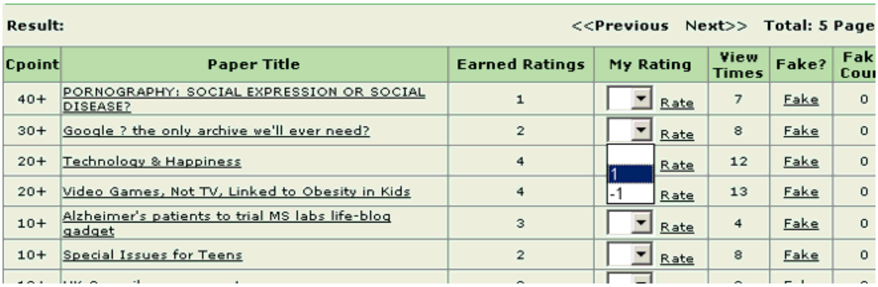
A challenge in systems that rely on decentralized moderation is to ensure that there are enough user ratings. MovieLens tried to motivate users to rate movies by sending them email-invitations [13]. The results showed that users seemed to be influenced more by personalized messages emphasizing the uniqueness of their contributions and by messages that state a clear goal (e.g. number of movies the user should rate). While this approach is questionable as a long-term solution because the effect of receiving email will likely wear off, it is interesting that personalization seems important and that setting specific goals are more persuasive than general appeals. To stimulate users to rate resources constantly, persistent incentives are necessary.

Our previous case study showed that different people had different contribution patterns. Some contribute many, but average (or even poor-quality) resources, while some contribute few, but very good ones. An adaptive motivational mechanism should encourage the users of the second category to contribute more resources unless the quality of their contributions starts to drop and inhibit the contributions from the users of the first category unless the users improve the quality of their contributions. The motivational mechanism should make users regard the quality and the quantity of their contributions equally.

Based on the discussion above, a collaborative rating system is introduced into the Comtella system, through which users can rate the resources in the community. The adaptive reward mechanism is designed based on the quality data from user ratings.

2. Collaborative rating

The Comtella rating mechanism is inspired from the Slashdot moderation system. In order to have a broader source of ratings, all the users can rate others' contributions by awarding them points (either +1 or -1). However, the users with higher membership levels receive more points to give out and are thus more influential in the community. To ensure that contributions have an equal chance to be read and rated initially, the initial rating for every new contribution is zero regardless of its providers' membership level or the quality of his/her previous contributions. In the end, the final rating for the contribution is the sum of all the ratings it has obtained. The summative rating for each contribution is displayed in the list of search results (Fig.1).



Result: <<Previous Next>> Total: 5 Page

Cpoint	Paper Title	Earned Ratings	My Rating	View Times	Fake?	Fak Cou
40+	PORNOGRAPHY: SOCIAL EXPRESSION OR SOCIAL DISEASE?	1	Rate	7	Fake	0
30+	Google ? the only archive we'll ever need?	2	Rate	8	Fake	0
20+	Technology & Happiness	4	1 Rate	12	Fake	0
20+	Video Games, Not TV, Linked to Obesity in Kids	4	-1 Rate	13	Fake	0
10+	Alzheimer's patients to trial MS labs life-blog gadget	3	Rate	4	Fake	0
10+	Special Issues for Teens	2	Rate	8	Fake	0

Fig. 1. A segment of a search result list

As a persistent incentive for users to rate contributions, a virtual currency is introduced, called "*c-point*". Whenever a user rates an article, he/she is awarded a certain number of c-points, depending on his/her reputation of giving high-quality ratings. The user can use the earned *c-points* to increase the initial visibility of his/her postings in the search result list. Most users desire that their contributions appear in salient positions, e.g. in the first place or among the top 10, because in those positions they will have a better chance to be read and rated. The Comtella search facility displays all the contributions matching a query in a sorted list according to the number of *c-points* allocated by the contributors (Fig.1). Unlike the mechanism in Slashdot, this one allows the user flexibility to invest c-point in a particular posting.

3. Community model, individual model and adaptive rewards

The adaptive reward mechanism is introduced as an improvement of the mechanism of hierarchical memberships [5]. The basic idea is to adapt the rewards of particular forms of participation for individual users and displaying personalized messages to them depending on their current status and reputations and the current need of the community, thereby influencing and directing the users' behaviors of contributing.

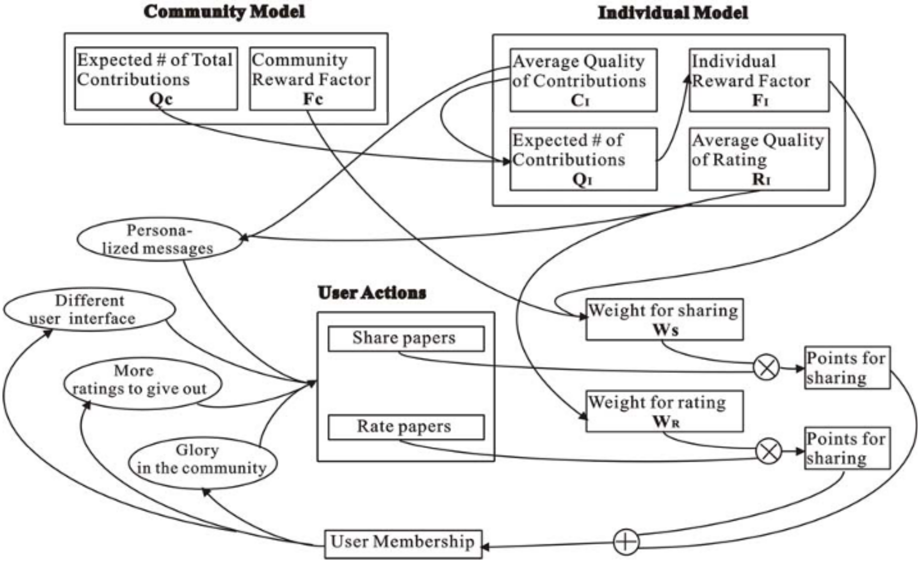


Fig. 2. An overview of adaptive reward motivation mechanism

Fig.2 presents an overview of the mechanism. The community model is used to describe the current phase of the whole community. It includes the expected sum of user contributions for current topic (Q_c) and the community reward factor (F_c). For each week, when a new discussion topic is introduced, Q_c is set by the community administrator (e.g. the instructor of the course) for the new topic, depending on his/her knowledge of certain features of the topic (e.g. how interesting it is expected to be for the users, how much materials are available) and the users' potential ability (e.g. how much time and energy they can devote, depending on their coursework, exams, etc.). F_c reflects the extent to which new contributions are useful for the whole community. Generally, new contributions are useful as soon as possible after a topic has been announced or opened. Therefore, F_c has its maximum value when a new topic discussion begins and decreases gradually with the time according to a function depicted in Fig.3.

Each user has an individual model that keeps statistical evaluations of his/her previous contributions and ratings and contains the data describing his/her current status. The average quality of a user's contributions (C_I) is defined in a straightforward way as the average summative rating of all the resources he/she has shared so far.

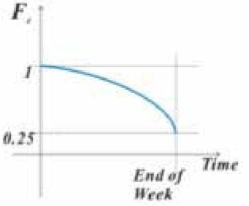


Fig. 3. The change of the community reward factor (F_c)

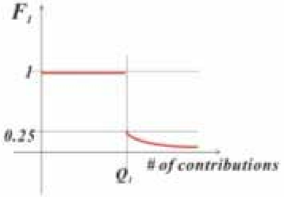


Fig. 4. The change of the individual reward factor (F_I)

However, the quality of user ratings can not be defined so easily, since they are by nature subjective. The average of all the ratings awarded to a given resource reflects the community criteria for quality and is more unbiased. Therefore, we chose to measure the quality of each rating for a given resource by the difference between the rating and the average rating that this resource has received so far. The quality equals to the reciprocal of the difference. Accordingly, the average quality of a user's ratings (R_I) equals to the average of the quality values of all the ratings he/she has made. Since this method can be skewed if users intentionally rate close to the average rating of the resource, the average rating should not be shown to the users directly.

The expected number of contributions of each user (Q_I) is a fraction of the total number of contributions that the community is expected to make for the topic, Q_c. The users with higher C_I will get a larger Q_I. If details are ignored, formula (1) can demonstrate how Q_c is distributed among users.

$$Q_I \approx Q_C \bullet \frac{C_I}{\sum C_I} \qquad \textbf{(1)}$$

The individual reward factor (F_I) defines the extent to which the user's contributions are being rewarded. F_I is a function that is a constant value as long as the number of the user's contributions is less than or equal to his/her Q_I. When the number exceeds the expectation, F_I drops to one fourth of the constant value instantaneously and keeps decreasing with the increment of the users' contributions (Fig.4)

Varying weights $W_i(t)$ for particular forms of participation are applied to compute the value of users' contributions and determine their membership levels, which are associated with different rewards and privileges. If we represent with $t=(1,2,3 \ldots T_i)$ the sequence of the contributions in each kind, the overall evaluation of a user's contributions (V) is calculated through formula (2).

$$V = \sum_{i=1}^{n}\left[\sum_{t=1}^{T_i} W_i(t)\right] \qquad \textbf{(2)}$$

The weights are adaptable to the states of the users' individual model and the community model at the current time. They, as well as the personalized messages, are conveyed to the users to influence their contribution patterns individually. The adaptive weight for sharing resources (W_S) is calculated through formula (3). Here W_{s0} is a constant, which is the initial value of the weight.

$$W_S = W_{S0} \bullet F_C \bullet F_I \qquad \textbf{(3)}$$

W_S is equal to W_{s0} when a new disucssion begins and the number of the user's contributions have not reached his/her expected value Q_I. After that, it decreases gently with time. Whenever the number of the user's contributions goes beyond his/her Q_I, W_s sharply decreases to one fourth of its original value and continues to decrease with the accumulation of the user's contributions and time.

It can be seen that W_S inherits the features of both reward factors, F_c and F_I. In this way, a user who shares many papers but does not pay enough regard to their quality gets a low C_I and a small Q_I and therefore, little reward for his/her subsequent contributions. Thus the personalized message to the user would be to contribute less in next period but improve the quality. This situation continues until the user finally improves his/her reputation for sharing. On the other hand, if a user tends to share a small number of good resources, he/she obtains a high C_I and a large Q_I. Potentially he/she will be able to earn more rewards by sharing more resources, and this continues until the quality of the user's contribution drops. For both kinds of users, early contributions always earn more points. Hence, W_S is able to restrict the quantity of user contributions, inhibit the contributions of poor quality, elicit good ones and stimulate users to share early in the discussion period.

The adaptive weight for giving ratings is proportional to the average quality of the users' previous ratings (R_I). The users who have gained a good reputation in making ratings get higher weight for their subsequent ratings, which stimulates them to rate more papers. However, those with poor R_I will not get much reward for rating articles. They have to rate less and improve the quality of their ratings to win their reputation back and this would be the suggestion of the personalized message.

4. Case study

To evaluate the effectiveness of the adaptive reward mechanism, a case study was launched in the course on "Ethics and Information Technology" offered by the Department of Computer Science at University of Saskatchewan in the second term 2004/2005 (Jan.-Apr. 2005). The study was carried out for eight weeks and the topic was updated weekly according to the curriculum. Thirty-two 4th-year students were the participants, who were encouraged to share web-articles related to the discussion topic using Comtella. The students were evenly divided into two groups: one group using the system with all the features of the proposed mechanism, including the functions of rating articles, earning and investing *c-points*, adaptive weights, personalized messages, etc. (test group / Comtella 1) and the other using the system with only the rating function (control group / Comtella 2). Since there might be some cultural and gender-based differences in the users' initial predisposition for participation, the assignment of users to groups was based on having equal proportion of Canadian to foreign and male to female students in each group. To avoid the effects that the contribution patterns of one group could have impact on the other group, the two groups inhabited two completely separated online communities, but shared the same classes, followed the same schedule, curriculum and coursework.

After the evaluation, post-experiment questionnaires were distributed to the participants to collect feedback about their experiences. The data from the questionnaires and the two systems were analyzed and contrasted to answer the following questions.

- Did the users in the test group (Comtella 1) give more ratings?

The data over the eight weeks suggested that the answer to this question was clearly positive since the number of ratings given in Comtella 1 was consistently (over each week) higher than that in Comtella 2. Throughout the eight weeks, the total number of ratings in Comtella 1 was 1065 and in Comtella 2 was 594. This clearly shows that the motivational mechanism with *c-points* and the associated rewards showed sustained effectiveness in stimulating users to rate articles.

- If more ratings was given in test group than in control group, did the summative ratings in test group reflect the quality of the contributions better?

Although we did not look into each article to evaluate its quality, we asked users about their attitude to the summative rating for their contributions. 56% of the users (9 users) in Comtella 1 felt that the final summative ratings could fairly reflect the quality of their contributions, while in Comtella 2, only 25% (4 users) thought so. This result shows that the increment of the quantity of user ratings can improve the accuracy of quality evaluation based on collaborative rating.

- Did the users in the test group tend to share resources earlier in the week?

According to the data collected in the eight weeks, the answer to this question is also positive. The users in Comtella 1 shared higher percentage of their contributions (71.3%) in the first three days of the week than the users in Comtella 2 did (60.6%) and the difference between the two groups was significant in each week (ranging between 7% and 14%).

- Did the users in the test group (Comtella 1) share the number of resources that was expected from them?

In the questionnaires, half of the users (8 users out 16) indicated they tended to share the number of resources that was expected from them. We calculated for each user the average difference between the actual shared number and the expected number over eight weeks and found that for half of the users the average difference was less than 2, which means these users contributed according to the expected number. Although the two groups of 8 users did not totally overlap, the results show that about half of the users were persuaded to share resources in or close to the number that was expected from them.

- Is there a significant difference with respect to the total number of contributions between the test and the control group?

The difference in the total number of contributions in the two groups is not significant (613 in Comtella 1 versus 587 in Comtella 2). The standard deviations of individual user contributions in the two systems are large, although in Comtella 1 it is slightly smaller than in Comtella 2 (30.18 versus 32.1). In Comtella 2 the top user is responsible for 21% of all the contributions, while the top user in Comtella 1 is responsible for 18% of the contributions. In both systems there was one user who didn't contribute at all.

- What is the user's perception with respect to cognitive overload and quality of contributions in each group?

Nine users in Comtella 1 and six users in Comtella 2 indicated in the questionnaire that they had to spend a lot of time time filtering out uninteresting posts, which means the effect of information overload emerged in both systems. As for the quality of the articles in both systems, we asked the users to give the rough percentages of the articles of high, medium and low quality in their own system. The data in Table 1 are the averages of users' estimations, which shows that their attitude towards the quality of the articles in their communities is basically neutral. It is hard to compare the degrees of informaiton overload and the quality of contributions in the two groups based on these data because the users in each group had experiences only in one system and there might have been ordering effects, in terms of different cognitive limits and criteria of quality evaluation among the students in the two groups. We plan to invite three experts to evaluate the articles in both systems to clarify their differences in terms of informaiton overload and the quality of contributions.

Table 1. Percentages of the articles of high, medium and low quality

Quality	**High**	**Mediun**	**Low**
Comtella 1	24.1%	46.3%	29.6%
Comtella 2	28.5%	42.3%	29.2%

5. Discussion and Conclusions

Designing incentives into the software to ensure that online communities are sustainable has been recognized as one of the most challenging and important problems in the area of social computing. We propose a dynamic, adaptive mechanism for rewarding contributions in an educational online community which takes into account the current needs of the community (e.g. more new contributions, versus more ratings, depending on the time since the topic is introduced and the current number of contributions) and the user's personal style of contributing (e.g. fewer but higher-quality contributions versus many mediocre ones). The hypothesis is that such a mechanism will stimulate users to contribute when and what is most useful for the community at the moment, thus achieving a level of activity that makes the community sustainable.

A study to test the effectiveness of the proposed mechanism was launched in a fourth-year undergraduate class with 32 students. Currently, the data collected from the participants are still being processed and analyzed. The results show that the mechanism encourages users to rate resources, motivates them to contribute early in the discussion and persuades at least half of them to contribute resouces in a specified number, thereby controling the amount of information in the community. More research is needed to find whether the quality of contributions improved. We are confident that the adaptive reward mechanism can improve the quality of contributions because it encourages the users who have a good reputation for sharing high-quality resources to share more and inhibit the contributions from the users who does not have a good repuation.

References

[1] J. Vassileva, J. Greer, G. McCalla, R. Deters, D. Zapata, C. Mudgal, S. Grant: A Multi-Agent Approach to the Design of Peer-Help Environments, in Proceedings of AIED'99, Le Mans, France, July, 1999, 38-45.

[2] J. Greer, G. McCalla, J. Vassileva, R. Deters, S. Bull and L. Kettel: Lessons Learned in Deploying a Multi-Agent Learning Support System: The I-Help Experience, in Proceedings of AIED'2001, San Antonio, 2001, 410-421.

[3] J. Vassileva, R. Cheng, L. Sun and W. Han: Stimulating User Participation in a File-Sharing P2P System Supporting University Classes, P2P Journal, July 2004.

[4] H. Bretzke and J. Vassileva: Motivating Cooperation in Peer to Peer Networks, User Modeling UM03, Johnstown, PA, 2003, Springer Verlag LNCS 2702, 218-227.

[5] R. Cheng and J. Vassileva: User Motivation and Persuasion Strategy for Peer-to-peer Communities, in Proceedings of HICSS'38 (Mini-track on Online Communities in the Digital Economy), Hawaii, 2005.

[6] P. S. Dodds, R. Muhamad and D. J. Watts: An experimental study of search in global social networks, Science 8 August 2003, 301: 827-829.

[7] S. R. Hiltz and M. Turoff: The network nation: Human communication via computer, Addison-Wesley Publishing Company, Inc., London, 1978.

[8] D. Shenk: Data smog: Surviving the information glut. HarperCollins, New York, 1997.

[9] Q. Jones and S. Rafaeli: User Population and User Contributions to Virtual Publics: A Systems Model, in Proceedings of the international ACM SIGGROUP conference on supporting group work, Phoenix, Arizona, 1999, 239-248.

[10] S. Johnson: Emergence: The Connected Lives of Ants, Brains, Cities, and Software, Publisher: Scribner, 2001, 152-162.

[11] C. Lampe and P. Resnick: Slash(dot) and Burn: Distributed Moderation in a Large Online Conversation Space, in Proceedings of CHI'2004, Vienna, Austria, Apr. 24–29, 2004.

[12] R. Merton and H. A. Zuckerman: The Matthew Effect in Science: the Reward and Communication Systems of Science are Considered, Science 199, 3810, 1968, 55-63.

[13] G. Beenen, K. Ling, X. Wang, K. Chang, D. Frankowski, P. Resnick and R. E. Kraut: Using Social Psychology to Motivate Contributions to Online Communities, in Proceedings of CSCW'04, Chicago, Illinois, Nov. 6–10, 2004.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

What Is The Student Referring To? Mapping Properties and Concepts in Students' Systems of Physics Equations

C.W. Liew [a,1] Joel A. Shapiro [b] and D.E. Smith [c]
[a] *Department of Computer Science, Lafayette College*
[b] *Department of Physics and Astronomy, Rutgers University*
[c] *Department of Computer Science, Rutgers University*

Abstract.
An ITS dealing with students' algebraic solutions to Physics problems needs to map the student variables and equations onto the physical properties and constraints involved in a known correct solution. Only then can it determine the correctness and relevance of the student's answer. In earlier papers we described methods of determining the dimensions (the physical units) of student variables. This paper describes the second phase of this mapping, determining which specific physical quantity each variable refers to, and which part of the set of constraints imposed by physics principles each student equation incorporates. We show that knowledge of the dimensions of the variables can be used to greatly reduce the number of possible mappings.

Keywords. Mathematics and science education, Physics Algebraic Equations

1. Introduction

In introductory physics courses, students are often presented a physics situation and asked to identify the relevant physics principles and to instantiate them as a set of equations. An Intelligent Tutoring System (ITS) for physics must *understand* the student's variables and equations in order to generate useful feedback. It must determine the physics principle used in each equation and to which properties and objects each variable refers. This is difficult when (1) there are many possible ways to specify a correct answer, (2) there are many reasonable names for variables that represent properties (the first mass could be m, m_1 or $m1$), or (3) the student submits an incorrect answer.

This paper describes a technique that reasons about all components of a student's submission to determine a correct interpretation. The approach taken is to compare the student's submission to a recorded correct solution for the problem (*i.e.*, the *exemplar*). If the student submits a correct solution and that solution is, equation by equation and variable by variable, a rephrasing of the the exemplar, the solution can be validated by identifying the mapping between the student's and exemplar's variables and equations. The number of possible mappings can be very large; however, the complexity of the

[1] Correspondence to: C.W. Liew, Department of Computer Science, Lafayette College, Easton PA 18042 E-mail: *liew@cs.lafayette.edu*, Phone: 1+(610)330-5537

search can be effectively managed when the dimensions of the student variables are known or can be determined.

Experience has shown that even correct answers seldom have a simple correspondence to an exemplar. Submissions that look very similar to an exemplar can be symptomatic of a misunderstanding of physics while those that look very different can be seen as correct once the concepts represented by the variables and equations are understood.

Consider a problem based on Atwood's machine, a frictionless pulley with two masses, m_1 and m_2 hanging at either end. A simplified[1] exemplar solution consists of

$$T_1 - m_1 * g = m_1 * a_1 \tag{1}$$

$$T_2 - m_2 * g = m_2 * a_2 \tag{2}$$

$$T_1 = T_2 \tag{3}$$

$$a_1 = -a_2 \tag{4}$$

Table 1 shows three possible submissions to the problem. The exemplar contains four equations but each of the three submissions contains at most three equations. Submission A is an incorrect solution that can result from a misunderstanding of how the direction of a vector effects a result. Submission B is a correct solution and can be derived by algebraic simplification of the exemplar. Submission C introduces a new variable that is not found in the exemplar. It cannot be derived by an algebraic simplification of the exemplar, but it is correct if M is understood to represent $m_1 + m_2$.

Submission A	Submission B	Submission C
$T - m_1 * g = m_1 * a_1$ $T - m_2 * g = m_2 * a_2$ $a_1 = a_2$	$T - m_1 * g = m_1 * a$ $T - m_2 * g = -m_2 * a$	$a = (m_1 - m_2) * g/M$

Table 1. Several Possible Submissions for Atwood's Machine

Previous approaches have either (1) severely constrained the student input to use prespecified variable names[5], or (2) used *strong scaffolding* to force the student to define the referents of her variables[7], or (3) used heuristic techniques to map the variables and equations[4]. Our algorithm considers all possible mappings of the student's variables and equations onto the exemplar, and computes the *distance* between the image and possible algebraic reductions of the exemplar set. If that fails to give a full match, equations are dropped from the student and exemplar sets to find the best mappings. The selected mappings are used to evaluate the submission for correctness and to identify possible errors.

2. Algebraic Physics Problems

An ITS for physics must first determine (a) what physics property (*e.g.* force, momentum) each variable represents and (b) to which object or system the property applies and at what time. Only then can the ITS determine if (c) each equation is relevant and correct and finally (d) if the set of equations is correct and complete. Some ITS's like ANDES [8,7] solve problems (a) and (b) by strong scaffolding that requires the student to define

[1] The full exemplar solution used in the experiment of section 4 contains 8 equations with 11 variables. In general, the exemplar should involve maximally primitive equations, those directly following from fundamental principles and the problem statement.

each variable, *i.e.* specify its dimensions and the object it applies to, before it is used. The system then uses its knowledge of the variables to determine the correctness of the equations using a technique called "color-by-numbers" [7,6]. In earlier papers [1,2,3] we described an alternative technique that determined the dimensions of students' variables from the context of the equations, thus solving issue (a). This paper describes our current work on solving issues (b), (c) and (d). We illustrate the problems involved with an example problem based on Atwood's machine, as shown in Figure 1a.

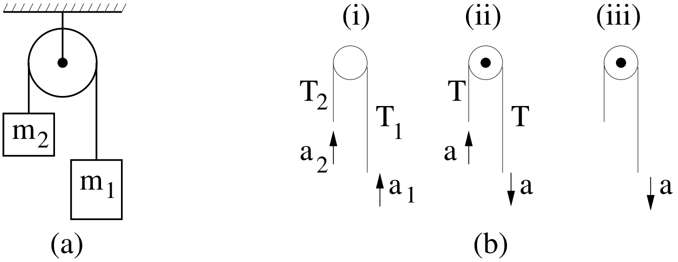


Figure 1. Atwoods Machine

A common problem using Atwood's machine asks for the equation(s) that would determine the acceleration of the mass m_1, assuming that m_1 and m_2 are not equal. Equations 1 through 4 represent a correct solution using variable set (i) in Figure 1b.

In an alternative formulation, the student chose to use a single variable a to represent acceleration and a single T for the tension. She implicitly used the principle that equates T_1 and T_2, and the constraint $a_1 = -a_2$, which comes from the fixed length of the cord. Variable set (ii) in Figure 1b identifies the variables used with such an approach. The resulting equations are "Submission B" in table 1.

In comparing the student's equations with the exemplar solution, an ITS must determine the mapping of the variables and equations from one set to the other. This process is complicated by several issues:

1. *variable renaming:* The student and the instructor may use different variable names to represent the same quantities. There is no restriction on the names of variables or choice of subscripts even though there are many standard variable names. There are also many commonly used variations, e.g. F, $Fnet$, F_1 can represent the same force.
2. *simple aliasing of one variable:* Frequently, variables that have the same magnitude and dimensions are aliased for one another. For example, the variables T_1 and T_2 in equations 1, 2 and 3 are equal to one another. In submission B of table 1, there is only a single variable T that is used to represent both, i.e., T is an alias for both T_1 and T_2.
3. *elimination by solution for variables:* There are many ways to specify the algebraic solution to a problem. These may involve using a greater or lesser number of variables and thereby a greater or lesser number of equations. For example, one very different but correct solution to the example problem is:

$$m_1 * g - m_1 * a = m_2 * g + m_2 * a$$

In this case, there is no variable representing the tension of the rope (commonly T, T_1 or T_2). Instead that variable has been solved for in one equation, which is eliminated from the set, and then substituted for in the other equations.

These issues result in there being many possible mappings between the variables and equations of a student's submission and that of the exemplar solution. Systems like ANDES [8,7] require that the student specify the mapping of variables. A mapping of equations (if it exists) can then be more easily derived. If the student input is not constrained in this way, the ITS must deal with the computational complexity issues. If each equation is evaluated singly, then each evaluation results in many possible interpretations and requires the use of strong heuristics to select a correct mapping [4]. Our algorithm considers all the variable and equation mappings simultaneously. The combination of all constraints greatly reduces the number of possible mappings that must be considered.

3. The Mapping Algorithm

The algorithm identifies properties and concepts by finding mappings of the variables and equations from a student set of equations to the variables and equations in an exemplar solution. The variables and equations in the exemplar are annotated with their dimensions and the associated physical principle [3].

The mappings of variables and equations are interdependent and the algorithm simultaneously finds a mapping for both variables and equations. This section describes how the dimensions of the variables are used to find the variable and equation mappings. Sections 3.1.2 and 3.3 show how the mappings can then be used to determine the algebraic differences between the student's equations and the exemplar.

3.1. Matching Dimensions

The dimensions of the variables are used to infer the dimensions of the equations. Each equation has a signature consisting of the dimensions of the equation and a vector of 1's and 0's, where a 1 indicates this equation contains the corresponding variable. Similarly, the signature of a variable consists of the dimensions of the variable and a vector of 1's and 0's, where a 1 indicates this variable is contained in the corresponding equation. The signatures are combined together to form a matrix where each row is the signature of an equation and each column is the signature of a variable (Table 2).

	T1	T2	m1	m2	a1	a2	g	dimension
Eqn 1	1	0	1	0	1	0	1	$\mathrm{kg} \cdot \mathrm{m/s^2}$
Eqn 2	0	1	0	1	0	1	1	$\mathrm{kg} \cdot \mathrm{m/s^2}$
Eqn 3	1	1	0	0	0	0	0	$\mathrm{kg} \cdot \mathrm{m/s^2}$
Eqn 4	0	0	0	0	1	1	0	$\mathrm{m/s^2}$
dimension:	$\frac{\mathrm{kg} \cdot \mathrm{m}}{\mathrm{s}^2}$	$\frac{\mathrm{kg} \cdot \mathrm{m}}{\mathrm{s}^2}$	kg	kg	$\frac{\mathrm{m}}{\mathrm{s}^2}$	$\frac{\mathrm{m}}{\mathrm{s}^2}$	$\frac{\mathrm{m}}{\mathrm{s}^2}$	

Table 2. Matrix of signatures for Equations 1 through 4

3.1.1. Comparison of Matrices

In this section we assume that the exemplar and the student set of equations have the same number of equations and variables of each dimensionality. A matrix of dimension signatures is constructed for both the solution set and the student set of equations. The goal is to find one or more correct mappings between the variables and equations of the two sets. A mapping between the two matrices is correct if the matrices are identical, *i.e.*

every entry in one matrix is identical to the corresponding entry in the other matrix and the *given*[2] variables are in the same columns in both matrices. When this happens, we have a *dimension map* between the student solution and the exemplar. Possible mappings are generated by permuting the rows and columns of the solution matrix subject to the following constraints:

- Rows (equation signatures) can be interchanged only if the equations have the same dimensions.
- Columns (variable signatures) can be interchanged only if the variables have the same dimensions.

In Table 2, if dimensions are ignored there are $4! \times 7!$ $(= 120,960)$ possible permutations. If we restrict row and column interchanges to those with the same dimensions then rows 1,2 and 3 can be permuted, columns 1 and 2 can be interchanged, columns 3 and 4 can be interchanged and columns 5, 6 and 7 can be permuted. The set of four equations (Equations 1 through 4) can yield 144 different permutations (mappings of variables and equations) that are dimensionally equivalent. We can further restrict the interchanges such that rows (equations) can only be interchanged if they use the same number of variables of each type. Applying this restriction to both row and column interchanges as well as constraining the given variables to be in the same columns in the exemplar and student matrices further reduces the number of permutations to 8. This technique when applied to the full exemplar solution for Atwood's machine (8 equations) reduces the number of permutations by a factor of 100 million from $8! \times 11! = 1.61$ trillion to $2! \times 4! \times 2! \times 2! = 9216$.

3.1.2. Evaluation of Equations for Correctness

The dimension information significantly reduces the search space but it is not sufficient to determine if the equations are correct. One of the many techniques for determining correctness, developed by Shapiro [6,7] and used in the ANDES system, is to instantiate the variables with random consistent values and then evaluate the equations to see if the values hold. This method, while effective for correct equations, does not help in identifying the causes of errors in equations. Our technique instead compares the mapped student equations with the corresponding equation from the solution set, term by term, to find the algebraic differences between the equations. This requires that the equations in both the solution set and the student set be represented in a canonical form as a sum of products. Our system can transform most equations which occur in introductory physics to this form. The algebraic differences (errors) that can be detected include (1) missing terms, (2) extra terms, (3) incorrect coefficients and (4) incorrect signs (a '$+$' instead of a '$-$' and vice versa). For this technique to be generally applicable and successful, it must also take into account differences that are not errors, such as various orderings of terms or factors, and multiplication of the entire equation by a constant. The algebraic differences are then used to identify the physics principles that have been incorrectly applied.

3.2. Dealing with Equation Sets with a Different Number of Equations/Variables

It is often the case that students will generate answers that contain a different number of variables through the use of algebraic transformations. The matching algorithm uses

[2] The *given* variables are those explicitly named in the problem presentation.

the exemplar solution to construct a lattice of equivalent sets of equations that contain a smaller number of equations and variables. Construction of the lattice proceeds as follows from the exemplar equations:

1. Initialize the lattice with the exemplar and mark it on the frontier.
2. The equations in each node on the frontier of the lattice are analyzed for variables that can be solved for in terms of other variables. Variables whose values are specified (givens) or that the student is supposed to find (the goal) are excluded.
3. Substitute for the variable in each of the other equations in the node. This results in a new set of equations with one fewer equation and forms a new node on the frontier in the lattice.
4. This process (steps 2 and 3) is repeated until the nodes on the frontier all contain only one equation for the goal variable.

The student's set of equations is then compared (Section 3.1.1) against the equations from nodes in the solution lattice that have the same number of equations and variables of each dimensionality. All valid mappings are collected into a list of possible mappings which are then used to evaluate the student's set for correctness (Section 3.1.2). If there is a mapping that results in the student's equations being evaluated as correct, then the student's equations are marked correct.

3.2.1. Application of Substitutions

Substitutions are applied only to the solution set of equations and not the student's set. This allows the system to refer to the student's original equations when generating feedback. In addition, this restriction greatly reduces the number of possible mappings. An exemplar set of 8 equations, if we ignore repetitions, results in a lattice that contains 2^8 nodes. This approach works only if the exemplar solution encompasses all the correct variations that the student might use. If the student uses a variable that is not in the solution set (*e.g.* submission C in Section 1), the algorithm will not be able to (a) find a map or reference for the variable (b) evaluate the equation for correctness.

3.3. Matching Incorrect or Incomplete Equation Sets

The algorithm has been extended to determine the mappings even when there are equations that are missing, extra, incorrect or irrelevant. This phase of the algorithm is executed when a complete dimension match of the variables and equations cannot be found.

Equations are systematically removed one at a time from the exemplar and/or the student set of equations. After removal of the non-matching equations, the matching algorithm (Section 3.1.1) can be used to match the remaining equations and variables. The variable maps that are found from the match can then be used to try to derive the complete variable maps.

The algorithm starts by taking each node in the lattice of correct solutions (Section 3.2). and making it the top of a new lattice where all the other nodes contain incomplete sets of equations with one or more missing equations. This results in many lattices with incomplete sets of equations except for the top of each lattice. A similar lattice of incomplete sets of equations is constructed for the student's set of equations. Starting from the top of the student lattice, the algorithm compares each node with the equivalent nodes (ones with the same number of equations and variables of each dimensionality) from the lattice of lattices created from the exemplar. The comparison stops after trying to match all nodes at a level in the student lattice if any dimension match is found (Section 3.1.1). These matches are then applied to the student's variables and equations to give a set that is evaluated for correctness (Section 3.1.2).

4. Experiments

We collected answers to four pulley problems from 88 students in an introductory physics course. One of the four problems was the Atwoods problem and our initial evaluation focused on the answers to that problem. The students were not restricted in any way except that they were asked to refrain from making algebraic simplifications to their answers. The student answers were evaluated against the exemplar with 8 equations and the results are described below.

- Five equation sets were dimensionally inconsistent.
- Three equation sets were dimensionally ambiguous. Dimensional ambiguities frequently arise when the students only enter a single equation. The single equation does not provide sufficient context for the system to uniquely determine the dimensions of one or more variables.
- 47 equation sets matched using substitutions and the matrix of dimensions. That is, they dimension-matched. These were further broken down into:
    - ∗ 31 equation sets that matched exactly when compared term by term with a node in the exemplar solution set lattice.
    - ∗ 16 equations sets that had algebraic differences consisting of either (1) an incorrect sign, (2) an extra term or (3) a missing term.
- 22 equation sets dimension-matched partially, *i.e.* only after elimination of one or more equations from either the student or solution set of equations. Six of the 22 equation sets had extra equations. The algorithm was able to identify the extraneous equations as well as determining that the remaining equations were both correct and complete.
- 11 sets of equations had no subset that dimension-matched any non-empty subset of equations in the exemplar set.

For comparison, we used the ANDES algebra subsystem [6] to evaluate the same set of equations. In this case, we had to define the variables explicitly before evaluating each set of equations. The ANDES system found the same results as our algorithm except for one instance where the student used a correct but non-standard formulation (submission (C) in Section 1). In this one case, applying substitution (Section 3.2) on the student set of equations would have resulted in the algorithm discovering that the answer was correct. ANDES would not have permitted the student to define or use the variable M.

4.1. Discussion

The results show that the algorithm performed as well as the ANDES system on the equation sets that both could solve. This indicates that the combination of our earlier algorithm for determining the dimensions of variables and this algorithm for matching equations and variables may be sufficient to relax the scaffolding, not requiring the student to explicitly define variables before using them. In addition, the algebraic differences detected will facilitate generation of specific and useful feedback to the student.

The technique is most successful when the student uses a larger number of equations, *i.e.* minimizes the use of algebraic simplifications. The additional equations provide a context that enables the technique to efficiently find the correct mapping of variables and equations in most instances. When a correct mapping can be found, the algorithm finds either one or two mappings and if there are two or more mappings, heuristics are used to select one. The algorithm has been shown to be effective on the example problem as it reduces the possible mappings to just one or two correct mappings.

The algorithm relies on the student using variables that can be mapped onto variables from the exemplar solution. This does not always happen, as in the case of submission (C) in Section 1. In those cases, we can apply the substitution algorithm to the student equations as well. This is applied as a last resort because (a) the number of possible matches grows very quickly and (b) it is difficult to generate reasonable feedback.

5. Conclusion

We have described a technique that determines the objects (and systems of objects) and properties that variables in algebraic equations refer to. The algorithm efficiently uses the dimensions of the variables to eliminate most of the possible mappings and find either one or two correct mappings which can then be further refined with heuristics. The technique is effective even if the student's answer uses a different number of variables and equations than the solution set. The mapping of variables and equations has been used to determine the algebraic differences between the student's answer and the solution set. This can lead to more effective feedback when the student's answer is incorrect. The technique has been evaluated on a small set of answers to one specific question and compares well with the results of a well-known system (ANDES) that uses much tighter scaffolding.

References

[1] LIEW, C., SHAPIRO, J. A., AND SMITH, D. Identification of variables in model tracing tutors. In *Proceedings of 11th International Conference on AI in Education* (2003), IOS Press.

[2] LIEW, C., SHAPIRO, J. A., AND SMITH, D. Inferring the context for evaluating physics algebraic equations when the scaffolding is removed. In *Proceedings of Seventeenth International Florida AI Research Society Conference* (2004).

[3] LIEW, C., SHAPIRO, J. A., AND SMITH, D. Determining the dimensions of variables in physics algebraic equations. *International Journal of Artificial Intelligence Tools 14*, 1&2 (2005).

[4] LIEW, C. W., AND SMITH, D. E. Reasoning about systems of physics equations. In *Intelligent Tutoring Systems* (2002), Cerri, Gouarderes, and Paraguacu, Eds.

[5] http://www.masteringphysics.com/.

[6] SHAPIRO, J. A. An algebra subsystem for diagnosing students' input in a physics tutoring system. Submitted to International Journal of Artificial Intelligence in Education.

[7] VANLEHN, K., LYNCH, C., SCHULZE, K., SHAPIRO, J., SHELBY, R., TAYLOR, L., TREACY, D., WEINSTEIN, A., AND WINTERSGILL, M. The ANDES physics tutoring system: Lessons learned. under review by IJAIED, 2005.

[8] VANLEHN, K., LYNCH, C., TAYLOR, L., WEINSTEIN, A., SHELBY, R., SCHULZE, K., AND WINTERSGILL, M. Minimally invasive tutoring of complex physics problem solving. In *Intelligent Tutoring Systems* (2002), Cerri, Gouarderes, and Paraguacu, Eds., pp. 367–376.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

The Effects of a Pedagogical Agent in an Open Learning Environment

Geraldine CLAREBOUT & Jan ELEN
Center for Instructional Psychology and Technology
Katholieke Universiteit Leuven
Vesaliusstraat 2, 3000 Leuven, Belgium

Abstract. Multiple studies have reported beneficial effects of embedding pedagogical agents in learning environments [1]. Most of these studies relate to effects in well-structured learning environments. In contrast, this contribution investigates the effects of pedagogical agents in open learning environments. A group that receives advice from a pedagogical agent is compared to a no-agent group. Dependent variables were performance, score on a transfer test and the use of tools. While groups did not differ on the performance test, rather surprisingly, the no-agent group outperformed the agent group on the transfer test. The no-agent group also spends significantly more time on the information list than the agent group. Results suggest that the pedagogical agent made the environment even more complex, causing additional cognitive load for the participants.

1. Introduction

Multiple studies have reported beneficial effects of embedding pedagogical agents in learning environments [1]. Moreno, Mayer, and Lester[2] for instance showed that students working in an environment with a pedagogical agent performed better than students who received only text-based information. Strikingly, most of these studies are done with well-structured environments. In these well-structured environments the agent acts as a coach who mainly delivers domain specific information or information on how to solve the problem at hand. The learning goals within these environments pertain mainly to learning specific information or procedures.

Whether pedagogical agents can also be helpful in more open learning environments is discussed in this contribution. Open learning environments are environments that (a) confront learners with ill-structured problems that have no specific solution and (b) offer students tools that can be used to solve the problem [3]. Open learning environments are characterized by a large extent of learner control. Learners decide for themselves whether and when the use of tools would be beneficial for their learning. Unfortunately, research on tool use [4] indicates that students do not (adequately) use tools in learning environments. Students seem to lack the metacognitive skills needed to ascertain what is beneficial for their learning [5].

These findings confront instructional designers with a dilemma. Open learning environments are advocated to foster the acquisition of complex problem solving skills, while learners seem to experience problems to handle such environments. A possible solution might come from the introduction of pedagogical agents. Such agents may help learners to handle open learning environments by providing (metacognitive) advice on the use of tools. In other words, pedagogical agents may help learners by inducing better adapted self-regulating strategies. Lee and Lehman [6] have found initial indications that advice might indeed help students to benefit more from tools. Baed on a study in which students received advice on what tools to use when solving a problem, they report positive results of advisement on tool use. Bunt, Conati, Huggett, and Muldner [7] also report some preliminary evidence for the beneficial effect of advice on performance.

In the study reported here, the potential effect on tool use of pedagogical agents/advice in open learning environments was explored. In a quasi-experimental pre-test post-test study, tool use and learning results of a group with an agent and a group without an agent were compared. The agent group is hypothesized to outperform and use tools more frequently than the control group.

2. Methodology

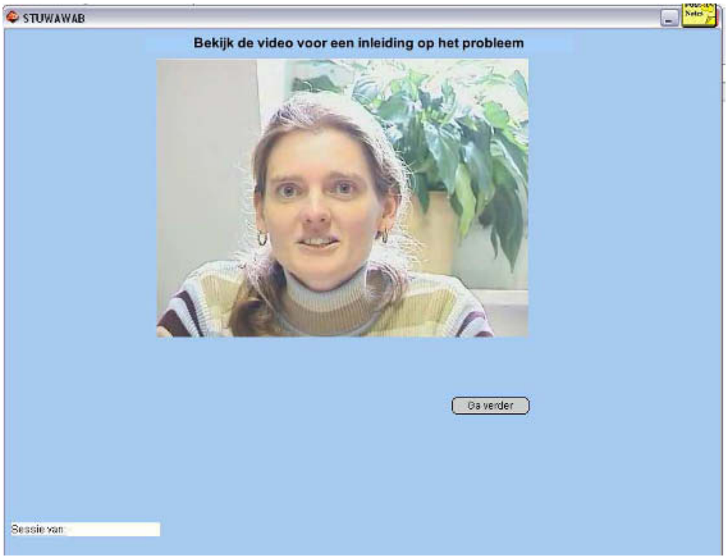
2.1 Participants

Twenty-eight first year educational sciences university students participated in the study. Participation was part of an introductory course on 'Learning and Instruction'. Fourteen served the experimental group (with agent) and fourteen served the control condition.

2.2 Materials

The computer-based learning environment was developed with Macromedia Director and is called STUWAWA (studying tool use with and without agents [8]). STUWAWA confronts students with an ecological problem. Participants are asked to select the most ecological drinking cup for use on a music festival, while also considering financial and security issues. Because students have to consider different perspectives and because there is no single right solution, the problem can be called ill-structured [9].

To make the task authentic, a real person (on video) introduces the problem. This person represents a member of the neighbourhood committee who requests help to solve the problem of garbage in their gardens after a music festival (see Figure 1).



Bekijk de video voor een inleiding op het probleem

Figure 1: Introduction of the problem

In order to solve the problem, participants have access to all sorts of information. In 8 short videos main actors present their view on the issue (environmental activist, festival participant, mayor, drank distributor, etc). Documents that relate to a specific actor's perspective can be accessed by clicking on a 'more information'-button. Non-categorised information can be accessed via an alphabetical list of documents. This list contains the titles of all documents available in the program. By clicking on a title participants get access to that specific document. In addition to information, s a diverse set of tools is put at students' disposal (information resources, cognitive tools, knowledge modelling tools, performance support tools, etc.). Table 1 provides an overview of the different tools, based on the categorization system for tools of Jonassen [10].

While working with the program participants can take notes in their personal workspace ("persoonlijke werkruimte" in Figure 2). Figure 2 presents a screen dump of the main screen of the program.

Table 1: Description of available tools in the environment

Icon	***Name***	***Kind of tool***	***Functionality***
opdracht	Assignment-tool	Information resource	Gives access to the explanation of the problem
video	Video assignment-tool	Information resource	Gives access to the introduction of the problem
	Information list	Information resource	Gives access to a list with all available information in the program
	Calculator	Performance support tool	Calculator (windows)
	Problem solving checklist tool	Knowledge modeling tool	Gives access to a problem solving help tool
	Reporting checklist-tool	Knowledge modeling tool	Gives access to help with reporting
	Technical support tool	Performance support tool	Technical help with the program
"Persoonlijke werkruimte"	Personal working space	Knowledge modeling tool	Students can take notes in this space, it is available during the whole problem solving process

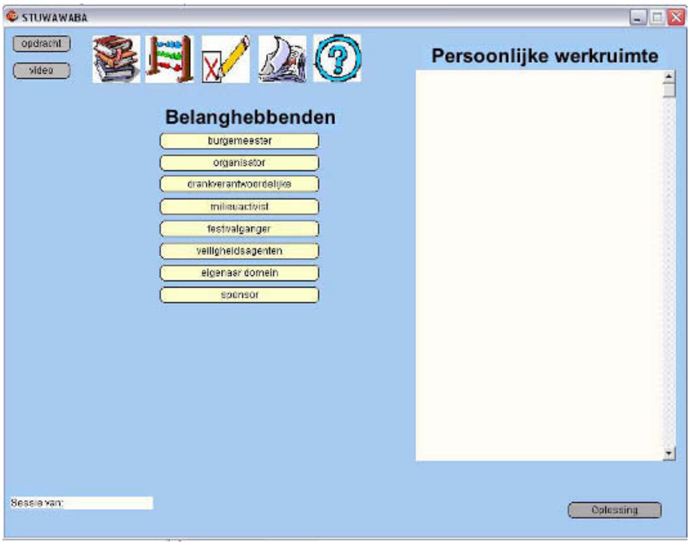


Figure 2: Problem solving environment

For solving the problem, the agent-condition participants receive problem-solving assistance from Merlin (Figure 3), a pedagogical agent (Microsoft Agent©). He directs a participant's attention towards the available tools. He explains the functionalities of the different tools and when students indicate to be willing to hand in their solution, he reminds them that they can consult their own notes at any time. Every five minutes, Merlin assesses what tools the student has already used. Based on this assessment he tells the students what tools have not yet been used. When a student has used (clicked on) all tools at least once, Merlin reminds the student that using tools might be helpful for solving the problem. Merlin always takes the initiative to deliver support himself, students cannot directly request support. Considering the results of a series of pilot studies [11], Merlin delivers support through means of on-screen text and personalised language.

Figure 3: Pedagogical agent[1]

To measure students' performance a pre- and post-test was administered. In the pre-test, students were requested what in their view was the optimal solution to the problem. Because students had not yet accessed the STUWAWA-environment, they could only rely on their prior knowledge to present their solution. After working in the problem solving environment, students were once more asked to present a solution (post-test). Students also received a transfer test. As for the regular test, a problem was introduced through means of a video-statement. This time the problem did not relate to drinking cups on a music festival but to vending machines in schools. In the video, a school director asked participants to identify the most ecological solution for a vending machine: drinking cans or bottles.

[1] Screen shot(s) reprinted by permission from Microsoft Corporation

All tests (pre-, post-, and transfer) were scored in an identical manner. Participants received one point for each argument provided, as well as for each counterargument. One point was subtracted for those arguments that contradicted participants' choice. Participants received one additional point for each perspective considered in their solution. An example is provided in Table 2.

Table 2: Illustration of arguments and scoring system

	Scoring	
Participant's answer	Perspective	Argument
Plastic cups: They do not need to be cleaned and as such no additional personel is needed which safes money.	1 (financial)	1 (personel)
They are safer than re-useable cups, although the waste to be cleaned up afterwards is larger than with re-usable plastic or glass cups.	2 (safety + organization)	- 1 (not safer than re-usable cups 1 (counter argument)
Total score: 4	3	1

Automatically generated log files of students' activities on STUWAWA were kept to gain insight in their tool use. Every tool click was registered in an Access database. In addition to the frequency of tool consultation, time spend per tool was logged as well.

Since it is assumed that students' metacognitive skills moderate the effect of advice, this variable was controlled for. Students' metacognitive skills were assessed with (a Dutch version of) the Learning Style Inventory (LSI) of Vermunt [12]. This questionnaire contains statements relating to students' self regulation or external regulation activities. For example, "I use the clues and goals provided by the teacher to know exactly what I have to do", or "If I don't understand a particular piece of study material, I search for other literature on this topic". Students were asked to indicate the extent to which they agreed with these statements on a six-point linkert-type scale, from totally disagree to totally agree.

Finally, a selection of students (n = 6) was asked to think aloud while solving the problem to gain more insight in their problem solving process.

2.3 Procedure

During the first lesson of the course on Learning and Instruction the LSI was administered. Students were then randomly assigned to the agent or the control group. In a second session (not part of regular course hours), students were asked to work in the environment. While the two groups were physically placed in separate rooms, all participants received the same introduction. They were asked to solve a complex problem. Furthermore they were told that in the environment some tools were available to assist them in their problem solving activities. The agent was not mentioned. Students could work on the problem at their own pace for maximum one hour. After one hour they were explicitly asked to submit their solution to the problem and to solve the transfer test.

The tests were independently scored by two researchers and interscorer reliability was calculated.

For the LSI, the reliability (Cronbach alpha) of the three scales of the inventory were calculated, namely the self-regulation, external regulation and no-regulation scale.

The log files were analyzed for the frequency of tool consultation and time spend on the different tools.

In order to determine the influence of the pedagogical agent on students' performance, a repeated measurement analysis was performed with condition as between subjects variable, the scores on the learning style inventory as co-variable and the pre- and post-test scores as within subjects variable.

For the influence on the transfer test and on frequency of tool use, ANCOVA's were performed with condition as independent variable, metacognitive skills as co-variable and the score on the transfer test, respectively tool use, as dependent variable.

3. Results

3.1 Reliabilities

Interscorer reliabilities for pre-,post- and transfer test varied in a first round between .864 and .990. After discussion between the two scorers, 100% agreement was reached.

Results of the LSI, revealed good reliabilities for the self-regulation scale, namely .866. For the other two scales, external regulation and no regulation, reliabilities of .516 and .694 were found. Given these disappointing reliabilities, only results on the self-regulation scale were used in further analyses.

3.2 Performance

Results of the repeated measurement analysis reveal that students significantly perform better on the post-test ($F(1,25) = 89.57$; $p \leq .05$; $eta^2 = .78$), but that there is no influence of the agent's advice. The two groups do not differ with respect to their learning gain. The learning gain (difference between the pre- and the post-test) was equal for both groups.

For the transfer test, however, ANCOVA reveals a significant difference between the two conditions ($F(1,26) = 6.46$; $p \leq .05$; $eta^2 = .21$) with the control condition outperforming the agent condition ($M_{co} = 5.25$; $SD_{co} = 2.17$; $M_A = 3.53$; $SD_A = 1.30$). Students in the control conditions gave significantly more arguments for their choice than students from the agent condition.

3.3 Tool use

An ANCOVA does not reveal any difference between the two groups for frequency of tool use. In both conditions, tools were consulted a similar number of times. For amount of time spend on a tool, a significant effect was found for the time spend on the information list ($F(1,27) = 4.26$; $p \leq .05$; $eta^2 = .14$). Students in the control condition spend more time on the information list ($M_{co} = .18$; $SD_{co} = .16$; $M_A = .09$; $SD_A = .16$).

4. Discussion and conclusion

The results of this study are surprising, to say the least. Overall, no effects of the pedagogical agent were found. Moreover, when an effect is found, this effect is the exact opposite of what could be expected. The agent seems to have a mathemathantic rather than a mathemagenic effect [5]; instead of facilitating learning, the agent seems to hamper learning Students who did not receive advice by the agent used the information list more frequently and performed better on a transfer test. The results suggest that the agent introduces more complexity in the environment and increases cognitive load [13]. This is in contradiction with the results of Lester, Converse, Kahler, Barlow, Stone and Bhoga [14] suggest that pedagogical agents do not increase cognitive load. It should be noted that the environment in which these authors tested pedagogical agents was more structured and that their claim is based on a comparison with different modalities of agents without a control group (no-agent).

Additional cognitive load might have been caused by the kind and timing of the agent's advice. The advice related purely to whether certain tools were used or not and was presented every five minutes irrespective of what the participants were actually doing. The thinking aloud protocols revealed that the advice frequency was too high. A five minute interval seemed to be too short. Given the functional equivalence of the advice presented and the high frequency of the advice, students started to simply ignore the agent. The students in the agent condition not only had to solve the problem and regulate the use of the tools, they also had to invest mental effort in actively ignoring the agent. Follow-up research will address this issue by (a) extending the time delay between two consecutive presentations of advice, (b) increasing the functionality of the advice by adapting it to the actual problem solving activities of the participants, and (c) looking into detail at what students actually do immediately after the advice is given.

An additional aspect that might have caused the observed lack of agent effect lays in the design of the environment. The stakeholders' videos may have been too powerful, partly because they are presented in the center of the screen. Given their visual power they may have attracted too much attention and time. Log files show that students systematically listen to all video-messages rather than actively looking for other (more-text-based but also more diversified) information, or using the other tools.

It should also be noted that the study performed here was done with a relatively small group of participants. This study does not allow for complex statistical analyses with more variables like students' ideas about the functionalities of the different tools, or students' motivation. In this study regulation skills were controlled for, although descriptive statistics showed that students hardly differ with respect to their regulation skills. A comparison of a more diverse group of students, a group with a high score on the regulation scale and a group with a low score might shed more light on this issue.

5. Acknowledgement

The authors want to thank Patrick Clarebout for his valuable programming work and Rebecca Léonard for her help with dataprocessing.

6. References

[1] Moreno, R. (2004). Animated pedagogical agents in educational technology. *Educational Technology, 44*(6), 23-30.

[2] Moreno, R., Mayer, R. E., & Lester, J. C. (2000). Life-like pedagogical agents in constructivist multimedia environments: Cognitive consequences of their interaction. In J. Bourdeau, & R. Heller (Eds.), *Proceedings of ED-MEDIA 2000. World conference on educational multimedia, hypermedia and telecommunications* (pp. 741-746). Charlotsville, VA: Association for the Advancement of Computers in Education.

[3] Hannafin, M., Land, S., & Oliver, K. (1999). Open learning environments: Foundations, methods and models. In C. M. Reigeluth (Ed.), *Instructional design theories and models. A new paradigm of instructional theory.* (Vol. 2, pp. 115-140). Mahwah, NJ: Lawrence Erlbaum.

[4] Clarebout, G., & Elen, J. (in press). Tool use in computer-based learning environments. Towards a research framework. *Computers in Human Behavior.*

[5] Clark, R. E. (1991). When teaching kills learning: Research on mathemathantics.

[6] Lee, Y. B., & Lehman, J. D. (1993). Instructional cueing in hypermedia: A study with active and passive learners. *Journal of Educational Multimedia and Hypermedia, 2*(1), 25-37.

[7] Bunt, A., Conati, C., Huggett, M., & Mulder, K. (2001). On improving the effectiveness of open learning environments through tailored support for exploration. In *Proceedings of the AIED2001.*

[8] Clarebout, G., & Elen, J. (2004). Studying tool use with and without agents. In L. Cantoni, & C. McLoughlin (Eds.), *Proceedings of ED-MEDIA 2004, World conference on educational multimedia, hypermedia and telecommunications* (pp. 747-752). Norfolk, VA: AACE.

[9] Spiro, R. J., Feltovich, P. J., Jacobson, M. J., & Coulson, R. L. (1991). Knowledge representation, content specification and the development of skills in situation-specific knowledge assembly: Some constructivist issues as they relate to cognitive flexibility. *Educational Technology, 31*(9), 22-25.

[10] Jonassen, D. H. (1999). Designing constructivist learning environments. In C. M. Reigluth (Ed.), *Instructional-design theories and models. A new paradigm of Instructional Theory, Volume II* (pp. 215-239). Mahwah, NJ: Lawrence Erlbaum Associates.

[11] Clarebout, G., & Elen, J. (2005, April). *In search of a modality and dialogue effec tin open learning environments*. Poster presented at the annual meeting of the AERA, Montréal, Canada.

[12] Vermunt, J. (1992). *Leerstijlen en sturen van leerprocessen in het hoger onderwijs. Naar procesgerichte instructie en zelfstandig denken.* [Learning styles and coaching learning processes in Higher Education]. Lisse: Swets & Zeitlinger.

[13]Chandler, P., & Sweller, J. (1991). Cognitive theory and the format of instruction. *Cognition and Instruction, 8,* 293-332.s

[14] Lester, J. C., Converse, S. A., Kahler, S. R., Barlow, S. T., Stone, B. A., & Bhoga, R. S. (1997). The persona effect: Affective impact of animated pedagogical agents. In *Proceedings of the CHI97 conference* (pp. 359-366). New York: ACM press

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Using Discussion Prompts to Scaffold Parent-Child Collaboration Around a Computer-Based Activity

Jeanette O'CONNOR, Lucinda KERAWALLA and Rosemary LUCKIN
Department of Informatics, University of Sussex,
Brighton, BN1 9QH, UK

Abstract: Research has shown that parents' collaborative involvement with their child's learning within the home context can be beneficial in improving their child's motivation and academic performance [1]. However, collaborative dialogue does not necessarily occur spontaneously because two users are sharing the same computer [2]. The current study focuses on the human-centred, iterative design and evaluation of a computer-based activity- Frankie's Fruitful Journey that utilises discussion prompts to scaffold parent-child collaboration around a weight and mass task within the home context. In the first design cycle, we identify when and where parent-child dyads could benefit from computer-based discussion prompts. In the second iteration we implement discussion prompts and report on their effectiveness in significantly increasing the quality of collaboration between parent and child. We conclude by discussing the future possibilities for the use of learner modelling techniques to support the provision of adaptive software scaffolding to guide parent interventions within parent-child dyads.

1. Introduction

There is a large body of evidence that suggests that parental collaborative involvement with their child's learning within the home context can be beneficial in improving their child's motivation and academic performance [1]. However, collaboration will not occur simply because a parent and child are working together on a single computer-based task [2]. We will discuss the iterative, human-centred development of Frankie's Fruitful Journey - software designed to engage parent and child in the collaborative completion of tasks within the domain of weight and mass. Our focus is on implementing and assessing the efficacy of system discussion prompts to guide collaboration and scaffolding within parent-child interactions.

1.1 Background

The research literature suggests that parents' involvement in their children's educational experience has a positive impact on student learning and motivation [1]. Students whose parents are involved in their schooling demonstrate higher academic achievement and cognitive development, which is particularly striking in cases where the parent works collaboratively with their child on school-work in the home setting (e.g. [3]). A significant body of research points to the educational benefits of working with more able partners (e.g. [4, 5]) and parents [6]. Vygotsky [4] argues that this is beneficial because a more able peer is able to encourage the child to work within their Zone of Proximal Development (ZPD).

He defines the ZPD as "the distance between the actual development level, as determined by independent problem solving, and the level of potential developments determined through problem solving under adult guidance or in collaboration with more able peers" ([4] p. 86). Studies have examined both the effects and the processes of effective parent-child interaction and the process of guided participation is one framework within which the term 'scaffolding' [6] can be conceptualised.

The scaffolding metaphor has become a lynchpin of Learner Centred Design which has promoted a design framework based upon socio-cultural philosophy. Software scaffolding has been informed by effective human interactions and benefited from the possibilities afforded by computing technology. All applications of software scaffolding aim to offer a means of enabling a learner or a group of learners to achieve success with a task beyond their independent ability. The way they have achieved this has varied with some placing emphasis upon the individual [7], some upon meta-cognition [8], and some upon collaboration [9].

Research looking at collaborative interactions whilst working at computers [10, 11] has concentrated upon the collaboration that occurs between learners, or between learners and teachers. As a result of these studies, Fisher [10] and Wegerif [11] have drawn up a taxonomy of three distinct types of talk: disputational talk, cumulative talk and exploratory talk. Disputational talk is characterised by disagreements and individualised decision-making, short assertions and counter-assertions. In cumulative talk, speakers build positively but uncritically on what the other has said, and characteristically repeat, confirm and elaborate utterances made by each other. In exploratory talk, partners engage critically but constructively with each other's ideas, offering justifications and alternative hypotheses.

It has been demonstrated that exploratory talk has most learning potential because it makes plans explicit, aids the decision making process and is used to interpret feedback [12]. It also makes knowledge publicly accountable and reasoning is more visible ([13] p.104). Mercer [13] has undertaken further research focusing upon the range of talk techniques that teachers use to guide children's learning. These include: Elicitations, Confirmations, Repetitions and Elaborations. He suggests that these techniques are being used by teachers to construct joint, shared versions of educational knowledge with their students. Whilst we agree that these techniques can be used to scaffold learners, it is unlikely that a classroom teacher will be able to provide the individualised intervention for all children that scaffolding necessitates. On the other hand, a parent working on an activity with a single child at home has the *opportunity* for one to one, sustained intervention but does not necessarily possess the appropriate skills. Therefore, we have identified a need to address the area of scaffolding parents to better enable them to assist their children.

Exploratory talk and guiding talk will not necessarily occur spontaneously because two users are sharing the same computer [2] and there is evidence of barriers at task and interface levels that can inhibit effective collaboration: the tendency for individuals to compete with each other [14], the adoption of turn taking behaviour at computers [15], difficulty recognising shared goals [16], and domination of the activity [15]. Kerawalla, Pearce, O'Connor, Luckin, Yuill and Harris [17] have attempted to address these problems with the design of their user interface - Separate Control of Shared Space (SCOSS). This provides each user with simultaneous, individual control over their own space on the screen which represents their progress on an identical task. SCOSS makes it necessary for both users to contribute to the task and in this way it ensures that there is the opportunity for equity at both task and input levels. The separate spaces also represent the current state of agreement or disagreement between the users and can be used to resource collaborative interactions and the agreement process. However, this research has found that whilst the SCOSS interface can effectively set the stage upon which collaborative interactions can occur, it does not mediate the *quality* of the discussions that take place. The current study has attempted to address this by introducing screen-based discussion prompts to the SCOSS interface, to scaffold collaborative interactions.

We discuss the iterative design of 'Frankie's Fruitful Journey' - educational software designed to encourage young children to think about weight and mass with a parent. The overall aim of the two studies presented here was to evaluate the utility of discussion prompts in this type of software environment, with a longer-term view to informing the design of an intelligent, flexible system. The first study focuses upon identifying both current levels of collaboration and parental scaffolding and identifies places where the dyads would benefit from prompts to assist them. The second study describes the development of discussion prompts and assesses their efficacy.

2. Frankie's Fruitful Journey

2.1 The learner-centred design process

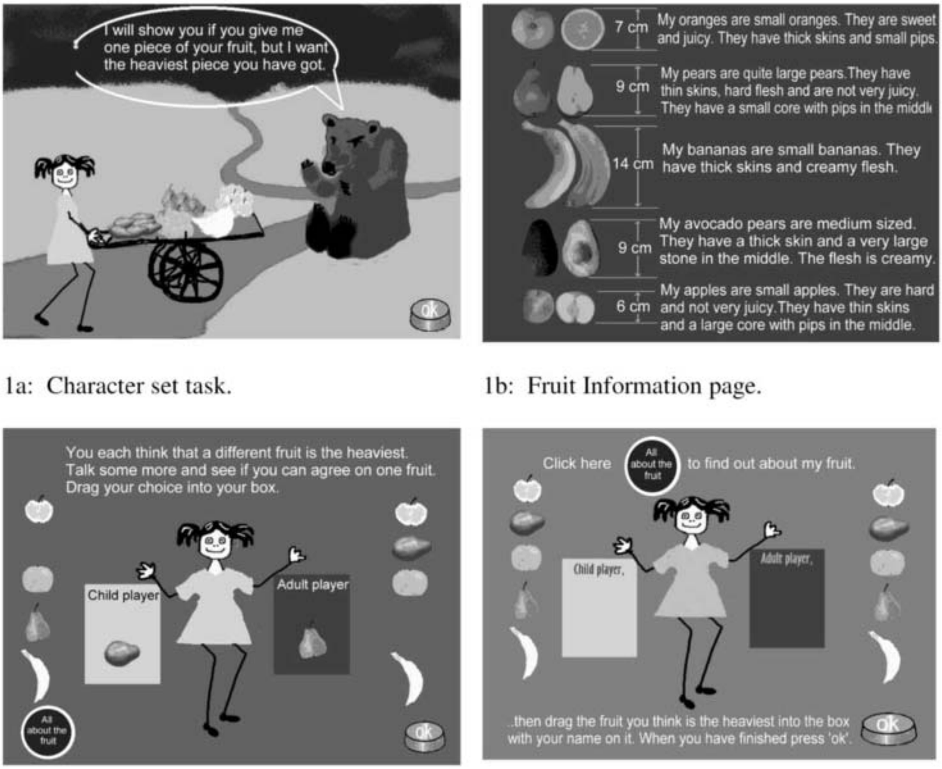
There is a growing recognition of the importance and value to be gained from involving users in the design process. For educational technology development this can include learners, teachers and parents. Work completed by Scaife and Rogers [18] and Robertson [19] has demonstrated the benefits as well as the challenges of the methodology, which confirms the topic as one that the AI community is addressing. The design process for Frankie's Fruitful Journey emphasises understanding the learning context, learner activities and artefacts, iterative design and evaluation cycles and low tech prototyping. The two iterative design cycles consisted of the design and evaluation of a paper prototype system, which then informed the re-design and evaluation of a second prototype developed using Macromedia Director MX. The first prototype design was informed by a range of requirements analysis activities. These included eleven contextual enquiry sessions in a primary classroom during which the computer skills of the children (aged 6 and 7 years) were assessed. They were also observed solving practical weighing problems. The approaches and resources used for teaching 'Mass' in the classroom were investigated through lesson observation and teacher interviews. The user-testing of the prototypes was carried out in the naturalistic home environments of the parents and children with all of the normal distractions (e.g. TV, siblings etc.).

2.2 Frankie's Fruitful Journey: the interface and the task

The interface and tasks in Frankie's Fruitful Journey were designed to encourage and enforce collaborative conversation by: establishing shared goals; providing both the parent and child with control of their own representation of the task; making visible the processes of agreement and disagreement; providing jointly accessible information resources and making a consensus of opinion necessary.

One task (of two) will be outlined here. The conversation prompts included in the second iteration are described later in section 2.4. In this task example, the users met a 'Big Old Bear' who would not tell them the way to a castle until they gave him the heaviest piece of fruit they had on their fruit cart (fig 1a). The task involved both the parent and child reading and discussing information relating to the size and composition of various fruits (fig 1b), and then deciding which fruit they thought was the heaviest. The assimilating and processing of the fruit information was a complex activity requiring textual comprehension, categorisations and the drawing of comparisons. It was a task that a child would struggle to complete without help from a more able partner and therefore presents an opportunity for parental scaffolding within the child's ZPD.

In accordance with the principles of Separate Control of a Shared Space (SCOSS) [14], both the parent and the child users were provided with a 'decision box' in which they placed their own choice of fruit (fig 1c), giving both the parent and child their own representation of the task. Once the individual decisions were made the dyads, they were then prompted to to reach an agreement (fig 1d).



I will show you if you give me one piece of your fruit, but I want the heaviest piece you have got.

1a: Character set task.

My oranges are small oranges. They are sweet and juicy. They have thick skins and small pips.

My pears are quite large pears.They have thin skins, hard flesh and are not very juicy. They have a small core with pips in the middl

My bananas are small bananas. They have thick skins and creamy flesh.

My avocado pears are medium sized. They have a thick skin and a very large stone in the middle. The flesh is creamy.

My apples are small apples. They are hard and not very juicy.They have thin skins and a large core with pips in the middle.

1b: Fruit Information page.

You each think that a different fruit is the heaviest. Talk some more and see if you can agree on one fruit. Drag your choice into your box.

1c: Individual representation of task.

Click here to find out about my fruit.

..then drag the fruit you think is the heaviest into the box with your name on it. When you have finished press 'ok'.

1d: Users prompted to reach agreement

Figures 1a-d: Frankie's Fruitful Journey task one

2.3 The first iteration: identifying the need for discussion prompts

The research aim of the first iterative cycle was to establish where and when parents needed more support to engage in exploratory and guiding talk. Ten volunteer parent and child (age 6 and 7 years) dyads worked face to face at the simulated computer in their home environments. The actions and the conversations of the participants were video recorded.

2.3.1 Analysis and findings

We developed a coding scheme that categorised conversational scaffolding utterances and exploratory talk. The categories of coding included: exploratory talk (explained opinion); unproductive talk (unexplained opinion) and a range of scaffolding categories taken from Mercer's [13] guiding talk techniques (e.g. elicitation of opinion). The video data was coded by hand then the data was transferred to a spreadsheet for analysis. The videos were watched several more times to clarify the context of some of the coded utterances and to closely examine the conversations associated with the processes of information assimilation and reaching agreement. The means and standard deviations for each of the utterance categories were calculated.

Analysis of the data revealed that interaction with Frankie's Fruitful Journey did yield *some* exploratory talk but that this was quite limited, with the parent simply changing their choices to match the child's choices and neither of them presenting 'arguments' to justify their decisions. These 'unexplained opinions' occurred more than twice as often in the conversations than 'explained opinions'. There were two good examples of conflict resolution conversations observed where both parent and child offered explanations of their opinions in response to each other's requests for justification. These requests were extremely effective examples of guiding talk, driving the productivity of the exchange and transforming cumulative talk utterances into exploratory talk exchanges.

2.4 The second iteration: incorporating discussion prompts

In light of the findings from the user testing during Iteration 1, Frankie's Fruitful Journey was adapted to incorporate conversation prompts to scaffold the collaborative decision-making process. The placing and content of these prompts was derived from the collation of the prompts parents were observed using during study 1. For each of the two tasks, there were two sets of conversation prompts: the first set encouraged the users to talk about the different characteristics of the fruit that might affect its weight, and the second set supported resolution of differences in opinion. All prompts were displayed on the screen that presented information about the fruits (table 1).

Table 1. Content and location of conversation points.

Information page access point.	**Conversation prompts**
Task 1: Make individual choices of heaviest fruit task	What things might make a fruit heavy? Will the biggest fruit be the heaviest? Will juicy or creamy fruit weigh more? Will thick skinned or thin-skinned fruit be heavier? Will pips in fruits make the fruit heavy?
Task 1: Agree on one choice of heaviest fruit.	Adult player say why you have made your choice of fruit. Child player say why you have made your choice of fruit.
Task 2: Make individual choices of weight order of fruits.	Can you remember which the heaviest fruit was? Will the smallest fruit be the lightest? Which shapes of fruit might be heavy and which shapes might be light? Will juicy or creamy fruits weigh less? Is the core of a fruit heavier or lighter than the flesh?
Task 2: Agree on a weight order of fruits.	Which fruits do you disagree about? Child player say why you think your choices are right. Adult player say why you think your choices are right.

The research aim was to investigate whether conversation prompts can facilitate exploratory and guiding talk in collaborative problem solving in parent-child dyads. Ten volunteer parent and child (age 6 and 7 years) dyads from two schools worked face to face at the computer in their home environments. Their actions and conversations were video recorded.

2.4.1 Analysis and findings

The video data was coded and analysed using the same techniques as for the first iteration. Mann-Whitney tests were performed to see if there were any significant differences between the total numbers of utterances made in each coding category that could be attributed to the inclusion of conversation prompts.

The inclusion of conversation prompts significantly increased the incidence of 'explained hypotheses' (exploratory talk) made by both the parents and children in both tasks (Table 2). This is an encouraging result indicating that the discussion prompts were effective in helping both the adult and the child to use more exploratory talk.

Table 2**:** Statistical analysis of changes in 'Explained hypothesis' utterances made by adults and children.

	Task	Mean utterances		Standard deviation		U value
		Study 1	Study 2	Study 1	Study 2	Study 1 and 2 compared
Child	Task 1	1.00	5.20	1.40	2.00	5*
	Task 2	1.60	6.20	2.20	1.90	9.5*
Adult	Task 1	1.10	3.50	1.40	1.30	12**
	Task 2	1.90	4.80	2.20	2.00	15.5**

*Significance level <0.01 ** Significance level <0.05

There was a significant increase in the quantity of child 'unexplained opinions' and in the quantity of adult 'elicitation of explanation' utterances. The first impression is that this is an undesirable outcome. However, on closer examination an interesting and important pattern emerges; the unproductive talk (unexplained opinion) by the child was transformed into a productive (exploratory) exchange by the parent. This suggests discussion prompts have taught parents when and where to intervene appropriately in the absence of prompts.

Table 3: Statistical analysis of changes in 'Unexplained opinion' utterances made by adults and children.

	Task	Mean utterances		Standard deviation		U value
		Study 1	Study 2	Study 1	Study 2	Study 1 and 2 compared
Child	Task 1	2.70	4.50	0.94	1.58	17.5*
	Task 2	4.20	5.80	1.90	1.70	22.5**
Adult	Task 1	2.60	1.50	0.9	1.17	21.5**
	Task 2	3.90	0.70	1.96	0.70	17.5*

* Significance level <0.01 ** Significance level <0.05

Prior to the inclusion of discussion prompts, many dyads were pre-occupied with a single property of the fruits that might affect weight (e.g. thickness of fruit skin). However, following the inclusion of discussion prompts, all dyads discussed all of the factors that might affect the weight of the fruits; they were made more aware of all of the characteristics of the fruit that could they could use to resource their joint understanding.

3. Discussion

This research has shown that the inclusion of computer-based discussion prompts significantly increased the utterances of exploratory talk because they reminded all parents to make their reasoning processes explicit. They also scaffolded the dyads' understanding of the type of information that was useful in the decision making process. Furthermore the system successfully scaffolded the parents to recognise where and when they could autonomously provide their own guiding prompts. These are encouraging results and represent the first step in understanding the significant role conversation prompts could play in enhancing collaboration and scaffolding within parents-child interactions.

We would like to build upon these finding in future by exploring the effects of varying the content, timing and wording of conversation prompts and then investigating the possibilities of using the SCOSS interface to capture data about individual roles in the task process. This will mean that the system will be able to provide intelligent conversation prompts tailored to the needs of the collaborators. This research could explore how parents use the software information resources, and provide adaptive systemic support that scaffolds their use of these resources to inform their decisions.

References

[1] Dyson, A. and Robson, A. (1999*) School, Family, Community: Mapping School Iinclusion in the UK*, National Youth Agency, London

[2] Steiner, K. E. and Moher, T. G.(2002) Encouraging Task-Related Dialog in 2D and 3D Shared Narrative Workspaces. *Proceedings ACM Conference on Collaborative Virtual Environments (CVE '02)*, Bonn, Germany, 39-46.

[3] Hickman, C.W., Greenwood, G.E., and Miller, M.D. (1995, Spring) High school parent involvement: Relationships with achievement, grade level, SES, and gender. *Journal of Research and Development in Education, 28*(3), 125-134

[4] Vygotsky, L. (1978) *Mind in society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press.

[5] Webb, N.M. (1995) Constructive Activity and Learning in Collaborative Small Groups. *Journal of Educational Psychology*, 87(3), 406-423.

[6] Wood, D., Bruner, J. and Ross, G. (1976). The role of tutoring in problem solving. *Journal of child psychology and psychiatry*, 17, 89-100.

[7] Koedinger, K. R., J. R. Anderson, et al. (1997). "Intelligent tutoring goes to school in the big city." *International Journal of Artificial Intelligence in Education* 8: 30-53.

[8] Luckin, R. and L. Hammerton (2002). Getting to know me: helping learners understand their own learning needs through metacognitive scaffolding. *Intelligent Tutoring Systems*. S. A. Cerri, G. Gouarderes and F. Paranguaca (Eds) Berline, Springer-Verlag: 759-771.

[9] Jackson, S., J. Krajcik, et al. (1998). The Design of Guided Learner-Adaptable Scaffolding in Interactive Learning Environments. *Conference on Human Factors in Computing Systems*, Los Angeles, California, United States, ACM Press/Addison-Wesley Publishing Co. New York, NY, USA

[10] Fisher, E. (1992) Characteristics of children's talk at the computer and its relationship to the computer software. *Language and Education*, 7 (2), 187-215.

[11] Wegerif, R. & Mercer, N. (1997) Using computer-based text analysis to integrate quantitative and qualitative methods in the investigation of collaborative learning, *Language and Education*, 11(4), 271–287.

[12] Light, P., Littleton, K. Messer, D. & Joiner, R. (1994), Social and communicative processes in computer-based problem solving, *European Journal of Psychology of Education*, 9, 93-110.
[13] Mercer, N. (1995). *The Guided Construction of Knowledge: Talk amongst Teachers and Learners.* Multilingual Matters, Clevedon.
[14] Benford, S., Bederson, B., Åkesson, K.-P., Bayon, V., Druin, A., Hansson, P., Hourcade, J., Ingram, R., Neale, H., O'Malley, C., Simsarian, K., Stanton, D., Sundblad, Y., & Taxén, G.(2000). Designing Storytelling Technologies to Encourage Collaboration between Young Children. *Proceedings of ACM CHI 2000 Conference on Human Factors in Computing Systems,* 1, 556-563.
[15] Scott, S., Mandryk, D., Regan, L., Inkpen and Kori, M. (2002) Understanding Children's Interactions in Synchronous Shared Environments. *Proceedings of CSCL 2002*(Boulder CO, January 2002), ACM Press, 333-341.
[16] Roussos M, Johnston A, Moher T., Leigh J., Vasilakis, C. and Barnes, C. (1999) Learning and building together in an immersive virtual world. *Presence*, 8 (3) 247-263.
[17] Kerawalla L., Pearce D., O'Connor J., Luckin R., Yuill N. and Harris A. (2005), Setting the stage for collaborative interactions: exploration of Separate Control of Shared Space.AIED 2005.
[18] Scaife, M., & Rogers, Y. (1999) Kids as informants: Telling us what we didn't know or confirming what we knew already. A. Druin (Ed.) *The design of children's technology*. San Francisco, CA: Morgan Kaufmann.
[19] Robertson, J. (2002) "Experiences of Child Centred Design in the StoryStation Project" In Bekker, M. Markopoulos, P. and Kersten-Tsikalkina, M. (Ed.s). *Proceedings of Workshop on Interaction Design and Children*, Eindhoven.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Self-Regulation of Learning with Multiple Representations in Hypermedia

Jennifer CROMLEY, Roger AZEVEDO, and Evan OLSON
Department of Human Development and Cognition and Technology Laboratory,
University of Maryland, 3304 Benjamin Building, College Park, MD 20742, USA
jcromley@umd.edu, razevedo@umd.edu, eolson01@yahoo.com

Abstract. A body of research has demonstrated when multiple representations of content help students learn. Few studies, however, have used process measures to understand what different cognitive processes students enact when learning from different representations. We collected pretest, posttest, think-aloud, and video data from 21 undergraduate students learning about the human circulatory system using a hypermedia encyclopedia. We measured learning as a change in a participant's mental model of the circulatory system from pretest to posttest. Students who learned more tended to spend less time in Text. While viewing Text alone, amount of learning was most strongly associated with verbalizing a smaller proportion of Feeling of Knowing, Free Search, and Selecting a New Informational Source. For Text + Diagrams, the amount of learning was most strongly associated with verbalizing a larger proportion of Inference and Self-Questioning. For Animation, the only significant variable was Summarizing. When not using the hypermedia environment, the significant variables were Feeling of Knowing, Prior Knowledge Activation, and Taking Notes. We close with implications for designing hypermedia environments for learning about complex science topics.

Introduction

Hypermedia environments, by definition, present learners with multiple representations of content (e.g., text, tables, diagrams, video clips). Learning, however, is not always improved by including multiple representations, either in Computer-Based Learning Environments (CBLEs) or in paper text [1]. A number of studies have shown that learners have difficulty coordinating different representations of the same content (e.g., [2]). A body of research has demonstrated *when* multiple representations help students learn complex science topics. A series of studies by Ainsworth and colleagues [3,4,5]; Chandler, Cooper, Sweller, and colleagues (e.g., [6]) and Mayer and colleagues (e.g., [7,8,9]), together with other studies (see the special issue of *Learning and Instruction* [10]), suggest that learning is improved when illustrations highlight important information, authors and designers avoid distracting information, and modalities are combined in ways that do not overload working memory.

Few studies, however, have used process measures to understand how students learn from multiple representations, that is, what different cognitive processes they enact when learning from different representations. We feel that better understanding the cognitive processes involved in using different representations can offer important guidelines for the design of CBLEs [11]. We begin by reviewing the handful of studies that we were able to identify that have collected process data from participants using multiple representations. We considered studies of learning with CBLEs or paper text, across different domains, and using different theoretical frameworks. We then describe our research questions.

Using a self-explanation framework, Ainsworth and Loizou [3] compared undergraduate students learning about the circulatory system from either paper text or diagrams. They prompted both groups to self-explain while learning. Participants completed pretests and posttests that included matching terms with definitions and drawing the flow of blood through the body. Students who were given diagrams had significantly higher scores on the diagram and flow measures at posttest. The researchers conclude on the basis of the verbal protocols that diagram participants engaged in more self-explanation while learning.

Using a cognitive strategy approach, Moore and Scevak [12] collected think-aloud, written free recall, standardized comprehension, and answers to literal and inferential questions from 119 of the highest-skilled readers in 5th, 7th, and 9th grades reading text and a diagram in science text. Older students tended to use a larger variety of different strategies while learning, and more often coordinated text and diagram than did younger students.

From an Information Processing Theory perspective, Hegarty & Just [13] used eye tracking to study cognitive processes when learning about complex pulley systems in text-and-diagram format on a computer. The researchers found that subjects integrated reading the text with fixating on the diagram rather than reading the complete text first and then fixating on the diagram. The location of the interruption in the subjects' reading of text tended to be at the end of a clause or sentence.

Using an expert-novice paradigm, Kozma and Russell [14] had both professional and undergraduate student chemists sort different representations—animations, equations, graphs, and videotapes of chemical processes—and verbally explain why they had sorted them in the way they did. The representations could be viewed on a computer, and were depicted on cards which were sorted. Whereas novices tended to sort different representations together (e.g., several videos together), experts made more multiple-media groupings. As with experts in physics (e.g., [15]) experts tended to give explanations based on laws and principles, whereas student explanations tended to describe surface features of the problem (e.g., movement, color).

Using a cognitive strategy approach, Lewalter [15] had undergraduate students think aloud while learning about how stars bend light in three different computer-based formats: text only, static diagrams, and animated diagrams. While students in both diagram conditions learned significantly more than those in the text condition, the think-aloud protocols showed that the static and animated diagram groups used different learning strategies. Most verbalizations were restatements of the text with little paraphrasing. However, the animated diagram group did verbalize more feeling of knowing, while the static diagram group engaged in more planning.

In summary, researchers have in a few cases collected process data from participants using multiple representations, but in only two studies did participants use hypermedia. There is therefore a need to collect process data from students while they are learning using multiple representations in hypermedia environments.

We designed a research study to investigate the relationship of Self-Regulated Learning (SRL) strategies used while learning from different representations (Text, Text + Diagrams, Animation, and Not in Environment) to learn about the circulatory system from a hypermedia environment. We measured learning as a change in a participant's mental model of the circulatory system from pretest to posttest—based on Azevedo and Cromley [17] and Chi [18]. The research questions were:

1) Which SRL variables are used while learning from different representations in hypermedia?
2) For each of the four different representations, what is the relationship between learning and amount of time spent in the representation?
3) For each of the four different representations, what is the relationship between learning and proportion of use of SRL variables?

1. Method

1.1 Participants

Participants were 21 undergraduate students (19 women and 2 men) who received extra credit in their Educational Psychology course for their participation. Their mean age was 22.4 years and mean GPA was 3.3. Forty-eight percent (*n* = 11) were seniors, 52% (*n* = 10) were juniors. The students were non-biology majors and the pretest confirmed that all participants had average or little knowledge of the circulatory system (pretest *M* = 5.29, *SD* = 2.61; posttest *M* = 8.52, *SD* = 2.64).

1.2 Materials and Equipment

In this section, we describe the hypermedia environment, participant questionnaire, pretest and posttest measure, and recording equipment.

During the experimental phase, the participants used a hypermedia environment to learn about the circulatory system. During the training phase, learners were shown the three most relevant articles in the environment (i.e., circulatory system, blood, and heart), which contained multiple representations of information—text, static diagrams, photographs, and a digitized animation depicting the functioning of the heart. Of the three most relevant articles, the blood article was approximately 3,800 words long, had 7 sections, 8 sub-sections, 25 hyperlinks, and 6 illustrations. The heart article was approximately 10,000 words long, had 6 sections, 10 sub-sections, 58 hyperlinks, and 28 illustrations. The circulatory system article was approximately 3,100 words long, had 5 sections, 4 sub-sections, 24 hyperlinks, and 4 illustrations. During learning, participants were allowed to use all of the features incorporated in the environment, such as the search functions, hyperlinks, and multiple representations of information, and were allowed to navigate freely within the environment.

The paper-and-pencil materials consisted of a consent form, a participant questionnaire, a pretest and identical posttest. The pretest was constructed in consultation with a nurse practitioner who is also a faculty member at a school of nursing in a large mid-Atlantic university. The pretest consisted of a sheet on which students were asked to write everything they knew about the circulatory system, including the parts and their purposes, how they work individually and together, and how they support the healthy functioning of the human body. The posttest was identical to the pretest. During the learning session, all participant verbalizations were recorded on a tape recorder using a clip-on microphone and the computer screen and work area were recorded on a digital videotape.

1.3 Procedure

The first two authors tested participants individually. First, the participant questionnaire was handed out, and participants were given as much time as they wanted to complete it. Second, the pretest was handed out, and participants were given 10 minutes to complete it. Participants wrote their answers on the pretest and did not have access to any instructional materials. Third, the experimenter provided instructions for the learning task. The following instructions were read and presented to the participants in writing.

Participant instructions were: "You are being presented with a hypermedia environment, which contains textual information, static diagrams, and a digital animation of the circulatory system. We are trying to learn more about how students use hypermedia environments to learn about the circulatory system. Your task is to learn all you can about the circulatory system in

40 minutes. Make sure you learn about the different parts and their purpose, how they work both individually and together, and how they support the human body. We ask you to 'think aloud' continuously while you use the hypermedia environment to learn about the circulatory system. I'll be here in case anything goes wrong with the computer and the equipment. Please remember that it is very important to say everything that you are thinking while you are working on this task." Participants were provided with pen and paper with which they could take notes, although not all did so.

1.4 Data Analysis

In this section, we describe scoring the pretest/posttest, coding the think-aloud protocols, and interrater reliability for the coding.

To code the participants' mental models, we used a 12-model coding scheme developed by Azevedo and Cromley ([17]; based on Chi [18]) which represents the progression from no understanding to the most accurate understanding of the circulatory system: (1) no understanding, (2) basic global concepts, (3) basic global concepts with purpose, (4) basic single loop model, (5) single loop with purpose, (6) advanced single loop model, (7) single loop model with lungs, (8) advanced single loop model with lungs, (9) double loop concept, (10) basic double loop model, (11) detailed double loop model, and (12) advanced double loop model. The mental models accurately reflect biomedical knowledge provided by the nurse practitioner. A complete description of the necessary features for each mental model is available in [17, pp. 534-535]. The mental model "jump" was calculated by subtracting the pretest mental model from the postest mental model.

To code the learners' self-regulatory behavior, we began with the raw data: 827 minutes (13.8 hr) of audio and video tape recordings from the 21 participants, who gave extensive verbalizations while they learned about the circulatory system. During the first phase of data analysis, a graduate student transcribed the audio tapes and created a text file for each participant. This phase of the data analysis yielded 215 single-spaced pages (M = 10 pages per participant) with a total of 71,742 words (M = 3,416 words per participant). We used Azevedo and Cromley's [17] model of SRL for analyzing the participant's self-regulatory behavior. Their model is based on several recent models of SRL [19, 20, 21]. It includes key elements of these models (i.e., Winne's [20] and Pintrich's [19] formulation of self-regulation as a four-phase process) and extended these key elements to capture the major phases of self-regulation: Planning, Monitoring, Strategy Use, Task Difficulty and Demands, and Interest. See Table 2 for the specific codes for each phase; for definitions and examples of the codes, see Azevedo and Cromley [17, pp. 533-534]. We used Azevedo and Cromley's SRL model to re-segment the data from the previous data analysis phase. This phase of the data analysis yielded 1,533 segments (M = 73.0 per participant) with corresponding SRL variables. A graduate student coded the transcriptions by assigning each coded segment one of the SRL variables.

To code the videotapes, we viewed each time-stamped videotape along with its coded transcript. We recorded time spent in each representation with a stopwatch and noted on the transcript which representation was being used for each verbalization. We defined Text + Diagrams as text together with any diagram, so long as at least 10% of the diagram remained visible on the computer screen. We defined Not in Environment as any time the participant read his or her notes (or verbalized in response to reading those notes), subsequently added to those notes without looking back at the screen (similar to Cox and Brna's *External Representations* [22]), or read the task instructions.

Inter-rater reliability was established by recruiting and training a graduate student to use the description of the mental models developed by Azevedo and Cromley [17]. The graduate student was instructed to independently code all 42 selected protocols (pre- and posttest descriptions of the circulatory system from each participant) using the 12 mental models of the circulatory system. There was agreement on 37 out of a total of 42 student descriptions yielding a reliability coefficient of .88. Similarly, inter-rater reliability was established for the coding of the learners' self-regulated behavior by comparing the individual coding of the same graduate student, who was trained to use the coding scheme with that of one of the experimenters. She was instructed to independently code 7 randomly selected protocol segments (30% of the 1,533 coded segments with corresponding SRL variables). There was agreement on 458 out of 462 segments yielding a reliability coefficient of .98. Inconsistencies were resolved through discussion between the experimenters and the student.

2. Results

2.1 Descriptive Statistics

Descriptive statistics on time spent in the four representations are shown in Table 1. On average, participants spent the most time in Text + Diagram (with little variability) and the least time in Animation, but with great variability in all representations other than Text + Diagram.

2.2 Research Question 1—Which SRL variables are used while learning from different representations in hypermedia?

Participants verbalized fewer SRL variables in representations other than Text + Diagram. See Table 1 for the number of SRL variables verbalized; not all SRL variables could be verbalized in all representations, e.g., Control of Context could only be enacted in the hypermedia environment. See Table 2 for which specific SRL variables were verbalized in each representation.

Table 1. Descriptive Statistics for Time Spent in Representations and Number of SRL Variables Verbalized

Representation	**Time Mean (SD) in min**	**No. SRL Variables Verbalized (% of possible)**
Text + Diagram	19.03 (3.48)	30 (100%)
Not in Environment	9.00 (6.17)	19 (73%)
Text	8.62 (5.17)	26 (90%)
Animation	2.72 (1.82)	15 (56%)

2.3 Research Question 2—For each of the four different representations, what is the relationship between learning and amount of time spent in the representation?

We computed Spearman rank correlations between the amount of time spent in each representation and jump in mental models. These results indicate which representations are associated with a higher jump in mental models from pretest to posttest. Proportion of time in Text had the highest correlation and the only significant correlation with mental model jump (r_s [21] = -.47, p < .05). The other representations had smaller and non-significant correlations: Text + Diagram (r_s [21] = .30, p > .05), Not in Environment (r_s [21] = .17, p > .05), and Animation (r_s [21] = .18, p > .05). Participants who spent a higher proportion of time in Text only had lower mental model shifts. We hypothesize that Text is either not as instructive as the other representations, or is more confusing than the other representations.

2.4 Research Question 3—For each of the four different representations, what is the relationship between learning and proportion of use of SRL variables?

In order to correct for the different number of verbalizations per participant and the different amounts of time spent in each representation, we transformed the raw counts of verbalizations of each SRL variable in each representation. We then multiplied the proportion of verbalizations for each SRL variable times the proportion of time spent in each representation. Finally, we computed Spearman rank correlations between the transformed proportion of use of SRL variables and jump in mental models for each representation. Results are shown in Table 2.

Table 2. Spearman Rank Correlation Between Proportion of Use of Each SRL Variable and Mental Model Jump, for Each Type of Representation

Variable [Raw number of verbalizations]	*Text*	*Text+Diagram*	*Animation*	*NIE*
Planning				
Prior Knowledge Activation [78]	-.322	-.060	-.300[1]	**.447***
Planning [10]	—	-.179	—	.171
Recycle Goal in Working Memory [29]	.094[1]	-.261	—	-.005
Sub-Goals [40]	-.208	.210	.210[1]	.094[1]
Monitoring				
Feeling of Knowing [105]	**-.523***	.205	.300	**.545***
Judgment of Learning [70]	-.304	.170	.021	.152
Monitoring Progress Toward Goals [13]	-.232	-.057	—	.136
Identify Adequacy of Information [14]	-.273	.017	—	—
Self-Questioning [11]	.022	**.435***	—	.300[1]
Content Evaluation [58]	**-.420***	-.086	NA	-.375*[1]
Strategy Use				
Draw [23]	.094[1]	.216	.107	.292
Summarization [125]	-.347	.170	**.435***	—
Taking Notes [321]	-.188	.347	-.062	**.470***
Read Notes [77]	NA	NA	NA	.181
Knowledge Elaboration [14]	.008	.136	—	-.206[1]
Coordinating Informational Sources [42]	NA	.041	—	.360
Find Location in Environment [6]	.278	.041	—	NA
Selecting New Informational Source [50]	**-.513***	.257	.352	.300[1]
Goal-Directed Search [12]	.059	.266	NA	NA
Free Search [32]	**-.441***	-.255	NA	NA
Mnemonics [9]	—	.296	—	.094[1]
Inferences [29]	**.379***	**.392***	—	.371
Re-Reading [97]	-.089	.276	-.053	—
Memorization [5]	—	-.114	—	.094[1]
Task Difficulty and Demands				
Time and Effort Planning [19]	.181	-.169	-.300	-.188
Control of Context [186]	**-.438***	-.103	.040	NA
Help Seeking Behavior [7]	.045	-.071	.094	—
Expect Adequacy of Information [13]	NA	.101	.307	NA
Task Difficulty [14]	**-.435***	-.240	.300	—
Interest				
Interest Statement [28]	-.316	.131	.085	—

* $p < .10$, — Dashes indicate the SRL variable was not usedby any participants in that representation, NA indicates code was not possible in that representation, [1] indicates code was used by only one participant.

While viewing Text alone, amount of jump was significantly associated with verbalizing a smaller proportion of Feeling of Knowing (FOK), Free Search (FS), Selecting a New Informational Source (SNIS), Control of Context (COC), Task Difficulty (TD), Content Evaluation (CE), and with a larger proportion of Inference (INF). While viewing Text + Diagrams, amount of jump was significantly associated with verbalizing a larger proportion of Inferences and Self-Questioning (SQ). While viewing the Animation, amount of jump was significantly associated with verbalizing a larger proportion of Summarizing (SUM). And when not using the hypermedia environment, amount of jump was most strongly associated with verbalizing a larger proportion of Feeling of Knowing, Prior Knowledge Activation (PKA), and Taking Notes (TN).

Looking at the same codes across representations, PKA was positively associated with jumping when it was verbalized Not in Environment, but was negatively associated with jumping when verbalized in Text. FOK was likewise positively associated with jumping when it was verbalized Not in Environment, but was negatively associated with jumping when verbalized in Text (that is, participants appeared to have some false sense of understanding when in text). SQ was positively associated with jumping when it was verbalized in Text + Diagram, but not in the other representations. CE was negatively associated with jumping when it was verbalized in Text (that is, participants appeared to have some false sense of the content being irrelevant when in text). SUM was positively associated with jumping when it was verbalized in the Animation (participants rarely took notes while watching the animation), whereas Taking Notes was positively associated with jumping when it was verbalized Not in Environment (i.e., adding to already-existing notes). SNIS was negatively associated with jumping when it was verbalized in Text (in this context, switching to the Animation from Text), but was non-significant when it was verbalized in Text + Diagrams or Not in Environment. FS (skimming) was also negatively associated with jumping when it was verbalized in Text. Inferences were positively associated with jumping when verbalized in Text or Text + Diagram. COC (frequently using the back arrow or hyperlinks) and TD were negatively associated with jumping when they were verbalized in Text.

3. Implications for Research and Design of Computer-Based Learning Environments

Our findings suggest certain guidelines for the design of hypermedia environments (see also Brusilovsky [23]). When students are using Text alone, they generally should be encouraged to switch to a different representation. However, to the extent that Text alone contains valuable information, students should be encouraged to draw inferences. For example, after the student reads 1-2 paragraphs, the environment could display a question that requires the student to draw inference from just-read text. In Text + Diagrams, the environment should encourage students to draw inferences, and should also encourage self-questioning. One simple way to do this would be to ask the student to write a question; the quality of the question need not be graded or scored, but we hope that by asking students to write a question, we would encourage monitoring and control processes.

In Animation, students should be encouraged to summarize. In our current research, we have successfully used experimenter prompts to get students to summarize; this could easily be embedded in a CBLE. Finally, when Not in Environment, students should be encouraged to judge their Feeling of Knowing, engage in Prior Knowledge Activation, and Take Notes. In our current research [17], we have also successfully used experimenter prompts to get students to judge their Feeling of Knowing; this could easily be embedded in a CBLE. Also, before students move to a new section in the environment, they could be prompted to read over their notes, recall what they learned previously, and consider revising their notes.

Acknowledgments

This research was supported by funding from the National Science Foundation (REC#0133346) awarded to the second author. The authors would like the thank Fielding I. Winters for assistance with data collection.

References

[1] Goldman, S. (2003). Learning in complex domains: When and why do multiple representations help? *Learning and Instruction, 13*, 239-244

[2] Ainsworth, S., Wood, D.J. , & Bibby, P.A. (1996) *Co-ordinating Multiple Representations in Computer Based Learning Environments*, Proceedings of EUROAI-ED, Lisbon.

[3] Ainsworth, S.E & Th Loizou, A. (2003) The effects of self-explaining when learning with text or diagrams. *Cognitive Science, 27*, 669-681.

[4] de Jong, T., Ainsworth, S., Dobson, M., van der Hulst, A., Levonen, J., Reimann, P., Sime, J., van Someren, M., Spada, H. & Swaak, J. (1998) Acquiring knowledge in science and math: the use of multiple representations in technology based learning environments in Spada, H., Reimann, P. Bozhimen, & T. de Jong (Eds) *Learning with Multiple Representations* (pp. 9-40) Amsterdam: Elsevier Science.

[5] Van Labeke, N., & Ainsworth, S.E. (2002). Representational decisions when learning population dynamics with an instructional simulation. In S. A. Cerri & G. Gouardères & F. Paraguaçu (Eds.), *Intelligent Tutoring Systems* (pp. 831-840). Berlin: Springer-Verlag.

[6] Kalyuga, S., Chandler, P., & Sweller, J. (1999). Managing split-attention and redundancy in multimedia instruction. *Applied Cognitive Psychology, 13*(4), 351-371.

[7] Mayer, R. E., & Anderson, R. B. (1992). The instructive animation: Helping students build connections between words and pictures in multimedia learning. *Journal of Educational Psychology, 84*(4), 444-452.

[8] Moreno, R., & Mayer, R. E. (1999). Cognitive principles of multimedia learning: The role of modality and contiguity. *Journal of Educational Psychology, 91*(2), 358-368.

[9] Mayer, R. E., Heiser, J., & Lonn, S. (2001). Cognitive constraints on multimedia learning: When presenting more material results in less understanding. *Journal of Educational Psychology, 93*(1), 187-198.

[10] Schnotz, W., & Lowe, R. (2003). External and internal representations in multimedia learning. *Learning and Instruction, 13*, 117-123.

[11] Lajoie, S. P., & Azevedo, R. (in press). Teaching and learning in technology-rich environments. In P. Alexander & P. Winne (Eds.), *Handbook of educational psychology* (2nd Ed.). Mahwah, NJ: Erlbaum.

[12] Moore, P. J., & Scevak, J. J. (1997). Learning from texts and visual aids: A developmental perspective. *Journal of Research in Reading, 20*(3), 205-223.

[13] Hegarty, M., & Just, M. A. (1993). Constructing mental models of machines from text and diagrams. *Journal of Memory & Language, 32*(6), 717-742.

[14] Kozma, R. B., & Russell, J. (1997). Multimedia and understanding: Expert and novice responses to different representations of chemical phenomena. *Journal of Research in Science Teaching, 34*(9), 949-968.

[15] Chi, M. T. H., Feltovich, P. J., & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science, 5*, 121-152.

[16] Lewalter, D. (2003). Cognitive strategies for learning from static and dynamic visuals. *Learning and Instruction, 13*, 177-189.

[17] Azevedo, R., & Cromley, J. G., (2004). Does training on self-regulated learning facilitate students' learning with hypermedia? *Journal of Educational Psychology, 96*, 523-535.

[18] Chi, M. T. H. (2000). Self-explaining expository texts: The dual process of generating inferences and repairing mental models. In R. Glaser (Ed.), *Advances in Instructional Psychology* (pp. 161-238). Mahwah, NJ: Lawrence Erlbaum.

[19] Pintrich, P. (2000). The role of goal orientation in self-regulated learning. In M. Boekaerts, P. Pintrich, & M. Zeidner (Eds.), *Handbook of self-regulation* (pp. 452-502). San Diego, CA: Academic Press.

[20] Winne, P. (2001). Self-regulated learning viewed from models of information processing. In B. Zimmerman, & D. Schunk (Eds), *Self-regulated learning and academic achievement: Theoretical perspectives* (pp. 153-189). Mawah, NJ: Lawrence Erlbaum.

[21] Zimmerman, B. (2001). Theories of self-regulated learning and academic achievement: An overview and analysis. In B. Zimmerman, & D. Schunk (Eds), *Self-regulated learning and academic achievement: Theoretical perspectives* (pp. 1-38). Mawah, NJ: Lawrence Erlbaum.

[22] Cox, R. and Brna, P. (1995). Supporting the use of external representations in problem solving: the need for flexible learning environments. *Journal of Artificial Intelligence in Education, 6*(2/3), 239-302.

[23] Brusilovsky, P. (2001). Adaptive hypermedia. *User Modelling & User-Adapted Interaction, 11*, 87-110.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

An ITS for medical classification problem-solving: Effects of tutoring and representations

REBECCA CROWLEY, ELIZABETH LEGOWSKI, OLGA MEDVEDEVA, EUGENE TSEYTLIN, ELLEN ROH and DRAZEN JUKIC
Center for Pathology Informatics, and Departments of Pathology and Dermatology
University of Pittsburgh School of Medicine
crowleyrs@msx.upmc.edu

Abstract. We report on the first evaluation of SlideTutor – a cognitive tutor in a medical diagnostic domain. The study was designed to evaluate both the overall impact of the tutoring system on diagnostic performance, and the effect of using two alternative interfaces to the system. The case-structured interface provides a case-centric view of the task, and the knowledge-structured interface provides a knowledge-centric view of the task. The study showed a strong effect of tutoring on performance in both multiple choice and case diagnosis tests for all students. Performance gains were maintained at one week. There were no significant differences between interface conditions on performance measures. Students in the knowledge-structured interface condition showed a significant improvement in correlating their certainty to their performance, which was not true of the case-structured condition. Ratings on a survey of user acceptance were significantly higher for the knowledge-structured interface.

1. Introduction

There are many challenges in creating intelligent medical education systems, including the ill-structured nature of diagnostic tasks, unique knowledge representation requirements, and the absence of formal notations for problem solving in this domain. Despite the significant need for intelligent educational systems in diagnostic medicine, very few have been developed [1-5], and to our knowledge, none have been evaluated to determine whether they improve diagnostic performance.

Medical ITS may take predominantly case-based approaches, knowledge-based approaches, or a combination. GUIDON [1,6-8] was an explicitly knowledge-based tutoring system – relying on its pre-formulated and rule-based problem solution to structure the discussion with the student. Early work on GUIDON provided important insights into the unique requirements of knowledge representation for ITS design, including the importance of forward-directed use of data, top-down refinement strategies, etiological taxonomies, incorporation of rules for expressing implicit "world relations", and the need to reason about evidence-hypothesis connections. Later tutoring systems considered alternative approaches to the knowledge representation problem. In MR Tutor [2], Sharples and DuBoulay utilized statistical indices of image similarity to develop training systems in radiology. The tutor exploited differences in measurements of typicality or similarity to train clinicians to recognize the full breadth of presentations of a given entity. Students learned by example from a library of radiologic images that represented “closed worlds’ of entities that were hard to distinguish.

The emphasis on case-based versus knowledge-based approaches is a fundamental design choice that has repercussions across the entire system from knowledge representation to interface. Few systems have explicitly studied how these choices affect skill acquisition, metacognition, and student experience. In the real world, the diagnostic training of physicians is usually based on a synthesis of case-based and didactic (knowledge-based) training. Early on, medical students often learn the initial approach to diagnosis in problem-based learning (PBL) sessions. PBLs use an actual clinical scenario - for example, a patient with elevated lipid levels. Students must work together to develop associated knowledge and skills. As they work through a case, students generalize, developing a more cohesive and unified approach to a particular problem than can be learned from a single case. They often incorporate group research on topics related to the scenario – for example, the pathophysiology of hyperlipidemia. Later in training, more expert residents and fellows work-up and diagnose patients under the supervision of an attending physician. During daily 'work rounds' – the diagnostic workup is the subject of an ongoing dialogue between attending physicians, fellows, residents, and medical students. In both cases, the goal is to help physicians synthesize a cohesive and unifying framework for common diagnostic cases. Into this framework, more complex atypical cases can be later incorporated.

In this study, we describe our first evaluation of SlideTutor - a cognitive tutoring system in pathology. The purpose of this study was twofold: (1) to determine whether the system was associated with any improvement in diagnostic accuracy and reasoning; and (2) to explore the relative effects of two diagrammatic reasoning interfaces on diagnostic reasoning and accuracy, metacognition, and student acceptance. One interface emphasizes relationships within an individual case, and the other incorporates a unifying knowledge representation across all cases.

2. System description

SlideTutor [9] is a model-tracing intelligent tutoring system for teaching visual classification problem solving in surgical pathology – a medical sub-specialty in which diagnoses are rendered on microscopic slides. SlideTutor was created by adding virtual slide cases and domain ontologies to a general framework that we developed to teach visual classification problem solving [10,11]. The framework was informed by our cognitive task analysis in this domain [12]. Students examine virtual slides using multiple magnifications, point to particular areas in the slide and identify features, and specify feature qualities and their values. They make hypotheses and diagnoses based on these feature sets. The expert model of the tutor constructs a dynamic solution graph against which student actions are traced. All knowledge (domain and pedagogic) is maintained in ontologies and retrieved during construction of the dynamic solution graph. The architecture is agent-based and builds on methods designed for the Semantic Web[13]. A fundamental aspect of SlideTutor is that it uses both real cases and its canonical representation of knowledge to help students develop their diagnostic skills. The modular nature of the system allowed us to test the identical system using very different methods for representing the relationship of the case data to the knowledge-base.

The **case-structured interface** (Figure 1A) uses a diagrammatic reasoning palette that presents a case-centric view of the problem. When features and absent features are added by the student, they appear as square boxes containing their associated modifying qualities. Hypotheses appear as separate rounded boxes, and may be connected to features using support and refute links. Hypotheses may be moved into the Diagnoses area of the palette when a diagnosis can be made (dependent on the state of the expert and the student models). Only the features present in the actual case are represented, but any valid hypothesis can be added and tested. At the end of each case, the diagram shows the relationships present in this single case. These diagrams will be different for each case. The interface is fundamentally constructivist, because students are able to progress through the problem space in almost any order, but must construct any unifying diagnostic representation across cases on their own.

In contrast, the **knowledge-structured interface** (Figure 1B) uses a diagrammatic reasoning palette that presents a knowledge-centric view of the problem. The interface is algorithmic. Students see the diagnostic tree unfold as they work through the problem. Features and absent features appear as square boxes containing their associated modifying qualities. As features are added, they are connected to form a path toward the diagnostic goal. When students complete any level of the algorithm by correctly identifying and refining the feature, the tutor reifies all of the other possible choices at that level. The current path (all identified features) is shown in yellow to differentiate it from other paths leading to other goals. Hypotheses appear as separate rounded boxes. When students make a hypothesis, the tutor places the hypothesis in the appropriate position on the diagnostic tree. When the hypothesis fits with the current evidence it is shown connected to the current path. When the hypothesis does not fit with the current evidence, it is shown connected to other paths with the content of the associated features and qualities hidden as boxes containing '?' - indicating a subgoal that has not been completed. Students may request hints specific to these subgoals. A pointer is always present to provide a cue to the best-next-step. By the conclusion of problem solving the entire diagnostic tree is available for exploration. The knowledge-structured interface therefore expresses relationships between features and hypotheses both within and across cases. Students can use the tree to compare among cases. At the end of each case, the diagram shows the same algorithm, but highlights the pattern of the current case.

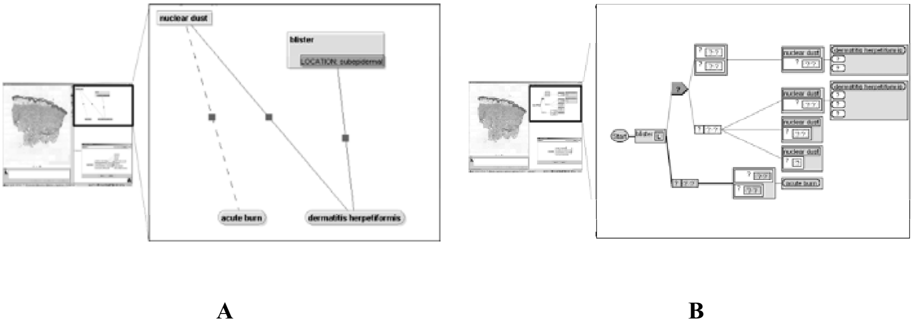


Figure 1 – Detailed view of the interactive diagrammatic palettes for case-structured (A) and knowledge-structured (B) interfaces. Both interfaces show the same problem state, in which nuclear dust and subepidermal blister have been identified as features, and acute burn and dermatitis herpetiformis have been asserted as hypotheses.

3. Research questions

- Is use of SlideTutor associated with improved diagnostic performance? If so, does interface representation affect performance gains?
- Does use of SlideTutor improve the ability of students to correctly gauge when they know or don't know the diagnosis. If so, does interface representation affect the accuracy of these judgments?
- Do students differ in their acceptance of these representations?

4. Methods

4.1 Design

Figure 2 depicts the between-subjects design. All subjects received the same pre-test, post-test, and retention test. On day one, subjects took the pre-test, were trained on the interface, worked for a fixed 4.5 hour period, took the post-test, and completed a user survey. During the working period, students worked with SlideTutor, and advancing at their own pace through twenty different dermatopathology cases. The sequence of cases was identical for all students.

Students who completed the cycle of cases, iterated again through the same sequence until the working period ended. One week later, they returned to complete the retention test. The entire study was performed under laboratory conditions.

4.2 Participants

Twenty-one pathology residents were recruited from two university training programs, and paid for their participation. Participants were assigned to use one of the two interfaces – eleven students used the case-structured interface and ten students used the knowledge-structured interface. Students were assigned to control for the number of years of training.

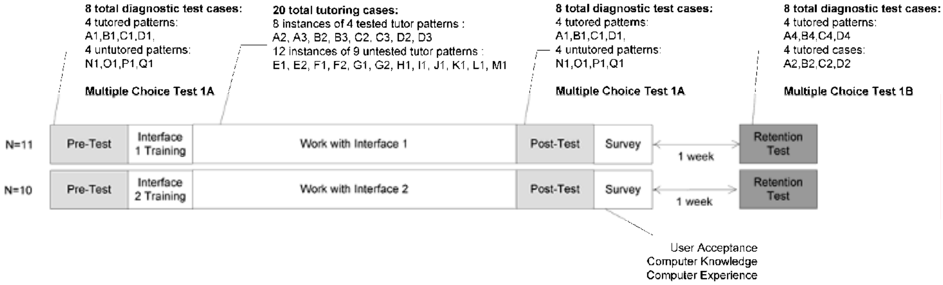


Figure 2: Study design with cases designated by pattern (A, B, C…) and instance (1,2,3…)

4.3 Participant survey

Students completed a three-section survey at the conclusion of the post-test. Section 1 contained 25 items, using a 4-point scale of agreement ("I agree with the statement: not at all | somewhat | moderately | strongly"). Question polarity varied. Items included statements regarding enjoyment, ease-of-use, trust in content, future-use, and comparison to alternative methods of study. Section 2 was a 17-item standardized instrument for measuring computer use [14]. Section 3 was a 20-item standardized instrument for measuring computer knowledge [14].

4.4 Assessments

All assessments were computer-based. Pre-test, post-test, and retention-test were identical in format, each consisting of two parts:

a) <u>Case diagnosis test</u> – subjects were presented with 8 different virtual slide cases using a virtual slide viewer but not within the tutoring system. For each case, students entered (1) diagnosis or differential diagnosis; (2) a justification for their diagnosis; (3) certainty about whether the diagnosis was correct on a 1-5 scale. We calculated a total case diagnosis score from (1) and (2), but also analyzed each component separately.
b) <u>Multiple choice section</u> – subjects answered 51 multiple choice and point-and-click questions that required them to locate features, identify features, indicate relationships of evidence to hypothesis, articulate differentiating features, and qualify features.

The pre-test and post-test (case diagnosis and multiples choice parts) contained identical questions. For the retention test, multiple choice questions were re-worded, and re-ordered. The case diagnosis part of the retention test did not overlap with the other tests. Students received no feedback on test-performance at any time.

4.5 Relationship of assessment content to tutoring session content

The knowledge representation of SlideTutor contains the relationships of features (evidence) and modifying qualities to diagnoses. A set of features and feature attributes constitutes a *pattern*. One pattern may lead to a single diagnosis, or to a set of diagnoses. One diagnosis may be associated with many patterns. Cases used in the working session reflected 12 different patterns – which we call *tutored patterns*. The actual cases that were used in the tutoring session are called *tutored instances*. The diagnostic sections of the pre-test, post-test and retention-test all contained four *tutored patterns*. The pattern was seen during tutoring, but the particular case in the test was not seen in the tutoring session. Pre-test and post-test also contained *untutored patterns* – cases that might have some features in common with the tutored patterns – but were associated with diagnoses not covered during the working session. The retention-test contained four tutored instances – actual cases that had been seen during the working period. Figure 2 depicts the relationship of cases used in assessments to cases used in tutoring session.

4.6 Analysis

Performance on pre-test, post-test and retention test was analyzed by MANOVA. We determined main effects and interactions for test and interface condition, including repeated contrasts. For performance-certainty correlations, slopes were computed using linear regression analysis, and were then compared by t-tests. The t-statistic was computed as the difference between the slopes divided by the standard error of the difference between the slopes. For participant surveys, we compared interface conditions using student's t-test. All analyses were performed in SPSS.

5. Results

Both conditions had a comparable mean level of training (20.2 months for the case-structured group, and 22.3 months for the knowledge-structured group). There were no significant differences between groups in the total number of cases completed during the working period. Eighteen of twenty-one (18/21) students completed the first cycle of twenty cases.

5.1 Learning outcomes

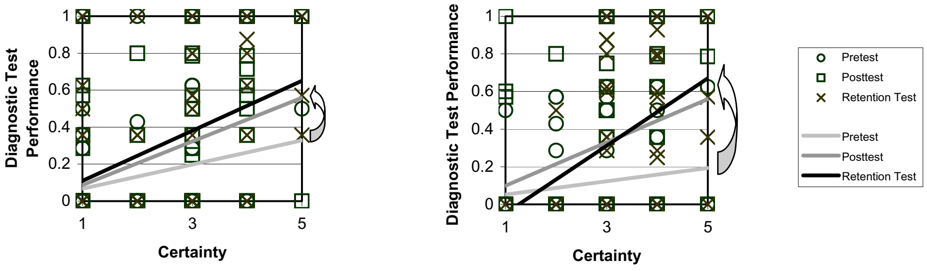
In both conditions, student performance was significantly higher at post-test (Table 1). This effect was observed in both the multiple choice and case-diagnosis tests. Scores on the multiple choice-test increased from 52.8 ± 12.5 % on the pre-test to 77.0 ± 10.4% on the post-test (MANOVA, effect of test, F=78.002, p<.001). In the case-diagnosis test, the effect was only seen in tutored patterns – where scores increased from 12.1 ± 8.7% on pre-test to 50.2 ± 22.6 % on the post-test (MANOVA, effect of test, F=64.008, p<.001). Case diagnosis scores are total scores reflecting both diagnosis and justification scores. Separate analysis of diagnostic accuracy and diagnostic reasoning scores are virtually identical to the aggregate scores shown in Table 1. No improvement was seen for untutored patterns. Performance gains were preserved at one week in both conditions, with no significant difference between retention test and post-test performance, for either multiple choice or case diagnosis tests. Notably, the case-diagnosis retention test contained completely different instances of the tutored patterns than those seen on the post-test and pre-test. Although overall performance improved across both groups, we did not observe a significant difference in performance gains or retention between the case-structured and knowledge-structured interfaces. Learning gains did not correlate with level of post-graduate training, computer knowledge or computer experience.

	Pre-test			Post-test			Retention test		
	Multiple Choice	Case Diagnosis		Multiple Choice	Case Diagnosis		Multiple Choice	Case Diagnosis	
Condition		Tutored Patterns	Untutored Patterns		Tutored Patterns	Untutored Patterns		Tutored Patterns	Tutored Cases
Combined	52.8 ± 12.5	12.1 ± 8.7	11.7 ± 7.9	77.0 ± 10.4	50.2 ± 22.6	10.4 ± 6.7	74.4 ± 13.3	47.8 ± 20.9	34.1 ± 17.0
Case	52.7 ± 12.6	11.8 ± 8.1	11.6 ± 8.9	75.5 ± 10.5	50.7 ± 25.3	7.8 ± 5.9	74.9 ± 13.5	44.6 ± 22.1	27.8 ± 18.5
Knowledge	52.8 ± 13.0	12.5 ± 9.8	11.9 ± 7.0	78.6 ± 10.5	49.6 ± 20.5	13.3 ± 6.7	73.8 ± 13.9	51.4 ± 20.1	40.9 ± 12.8

Table 1. Pre-test, post-test, and retention test scores (mean ± SD)

5.2 Correlations of performance to certainty

Analysis of certainty rating and performance on case diagnosis test questions show that students are relatively inaccurate in assessing their performance before tutoring. Performance-to-certainty ratios rise after tutoring as certainty becomes more accurately correlated with performance. For students using the knowledge-structured interface (Figure 3B) slope significantly changes from pre-test to post-test ($p<.05$) and from pre-test to retention test ($p<.01$). Slopes observed in pre-test, post-test, and retention-test were not significantly different for students who used the case-structured interface (Figure 3A). The difference between slope changes between the two conditions did not reach significance.



A

B

Figure 3. Correlations of performance to certainty during pre-test, post-test and retention test for (A) case-structured interface and (B) knowledge-structured interface.

5.3 Participant acceptance

Students using the knowledge-structured interface had a higher total survey score (82.6 + 8.8) when compared with students using the case-structured interface (72.6 + 7.168). The difference was significant by student t-test ($p <.05$). Students using the knowledge-structured interface rated the tutor higher in almost all categories, but especially in (1) how usable the system was; (2) how much they enjoyed using the system; and (3) how likely they were to use the system in the future.

6. Discussion

To our knowledge, this is the first study evaluating the effect of an intelligent tutoring system on medical diagnostic performance. Our results show a highly significant improvement in diagnostic skills after one tutoring session, with learning gains undiminished at one week. The selection of cases for the pre-test, post-test, working period, and retention-test were designed to mitigate well known problems in evaluating learning outcomes in medical domains. Pre-test and post-test were identical, and did not contain cases seen in the tutoring session. The equivalency of these tests is important because it is often extremely difficult to utilize multiple-form tests when dealing with real medical cases. Matching test forms to control for case difficulty is challenging, because it is often unclear what makes one case more or less difficult than another. Our demonstration of a strong increase in performance from pre-test to post-test cannot be explained by differences in the level of difficulty of non-equivalent tests. The absence of any improvement for the untutored patterns cases suggests that learning gains were also not related to re-testing.

We also used this evaluation to help us determine the kind of problem representation to use in our tutoring system. Unlike many domains in which model-tracing tutors have been used, medicine has no formal problem-solving notation. In particular, we wanted to determine whether two very different external representations would differ in terms of skill acquisition, metacognition, or user acceptance. Our results show increased acceptance of the more knowledge-centric knowledge-structured interface, but no significant difference between these interfaces for gains in diagnostic accuracy or reasoning. There was a trend toward increased performance-certainty correlation for students in the knowledge-structured condition compared to students in the case-structured condition. We expect to repeat the study examining potential meta-cognitive differences between the interfaces, using more subjects and a scale that permits finer discrimination.

Why is it important that students come to match their certainty to performance as closely as possible? When practitioners are uncertain about a diagnosis, they can seek consultation from an expert in the sub-domain or perform further diagnostic testing. Consultation is a particularly common practice in pathology subspecialties like dermatopathology. When practitioners are overly certain about diagnoses that turn out to be wrong, significant harm can be done because incorrect diagnoses are assigned without use of consultation or additional diagnostic procedures. On the other hand, diagnosticians who are overly uncertain may hinder the diagnostic process as well, by ordering unnecessary studies, and delaying diagnosis. An important part of developing expertise in diagnosis is learning to balance these two potential errors.

It could be argued that the case-structured interface provides a false sense of security, because students who use this interface have only the relationships within each case to use in judging their performance. For example, when they create a hypothesis for a particular pattern and get it right, they cannot see that there are many other similar, but slightly different, patterns that lead to other diagnoses. As with other cognitive tutors, the correct solution path is enforced - students may make errors on individual skills, but always come to the correct solution in the end. Case-centric representations may limit improvements in self-assessment because students never experience the diagnostic "near-misses." In contrast, the knowledge-structured interface provides a way to visualize the entire decision space at once, and lets students see the effect of subtle pattern differences on diagnosis across all cases. It also lets students see parts of the decision space that they have not been trained in. Knowledge-centric representations might support improvements in self-assessment because students can visualize diagnostic "near-misses" even though the enforced solution-path prevents them from experiencing them.

7. Future work

Extensive process measures were obtained during this study, which have not yet been analyzed. What parts of this task are difficult? How quickly do students reach mastery on skills? How predictive are student models of test outcomes? Are there differences between interfaces conditions in skill performance, time to mastery, or use of hints? Future work will address these questions.

Both of these interfaces have interesting properties that could be exploited in future work. The case-structured interface allows students to create their own diagrammatic 'argument' and is therefore amenable to manipulation of the feedback cycle. With this interface, we could implement a gradual relaxation of the 1:1 relationship of student action and tutor response that is typical for immediate-feedback in cognitive tutors. Our architecture permits cycles that evaluate the student's solution after a variable number of student actions. Tutor responses could be used to annotate the student's diagram in order to explain the "stacking" or errors that can occur when feedback is not 1:1.

In contrast, the knowledge-structured interface could be used to help students develop cohesive models of the diagnostic space. To date, our tutoring system provides feedback that relates to both the individual case and the knowledge base (diagnostic algorithm). But the unifying nature of the knowledge-structured interface could facilitate development of feedback that references other cases that have already been seen. For example, when students identify evidence or suggest relationships that were true in previous cases, but not in the current case – the diagnostic algorithm could be used to reference the veracity of the statement in the previous case, but point out how the current case differs. Also, the algorithm could be used to interactively revisit features or relationships in previous cases when students want to be reminded of their characteristics.

Acknowledgements

This work was supported by a grant from the National Library of Medicine (LM007891). This work was conducted using the Protégé resource, which is supported by grant LM007885 from the United States National Library of Medicine. SpaceTree was provided in collaboration with the Human-Computer Interaction Lab (HCIL) at the University of Maryland, College Park.

References

[1] Clancey WJ. Guidon. J Computer-based Instruction 10:8-14,1983.

[2] Sharples M, Jeffery NP, du Boulay B, Teather BA, Teather D, and du Boulay, G.H. Structured computer-based training in the interpretation of neuroradiological images. International Journal of Medical Informatics: 60: 263-280, 2000.

[3] Azevedo R, and Lajoie SP. The cognitive basis for the design of a mammography interpretation tutor. International Journal of Artificial Intelligence in Education. 9:32-44, 1998.

[4] Eliot CR, Williams KA and Woolf BP. An Intelligent Learning Environment for Advanced Cardiac Life Support. Proceedings of the AMIA Annual Fall Symposium, Washington, DC. 1996; 7-11.

[5] Smith PJ, Obradovich J, Heintz P, Guerlain S, Rudmann S, Strohm P, Smith JW, Svirbely J, and Sachs L. Successful use of an expert system to teach diagnostic reasoning for antibody identification. Proceedings of the Forth International Conference on Intelligent Tutoring Systems. San Antonio, Texas. 1998; 354-363.

[6] Clancey WJ. Knowledge-Based Tutoring - The GUIDON Program. Cambridge, MA: MIT Press, 1987

[7] Clancey WJ. Heuristic Classification. Artificial Intelligence 27:289-350, 1993.

[8] Clancey WJ and Letsinger R. NEOMYCIN: reconfiguring a rule-based expert system for application to teaching. Proceedings of the Seventh Intl Joint Conf on AI, Vancouver, BC. 1981; 829-835.

[9] Crowley RS, Medvedeva O and Jukic D. SlideTutor – A model-tracing Intelligent Tutoring System for teaching microscopic diagnosis. IOS Press: Proceedings of the 11th International Conference on Artificial Intelligence in Education. Sydney, Australia, 2003.

[10] Crowley RS and Medvedeva OP. A General Architecture for Intelligent Tutoring of Diagnostic Classification Problem Solving. Proc AMIA Symp, 2003: 185-189.

[11] Crowley RS, Medvedeva O. An intelligent tutoring system for visual classification problem solving. Artificial Intelligence in Medicine, 2005 (in press).

[12] Crowley RS, Naus GJ, Stewart J, and Friedman CP. Development of Visual Diagnostic Expertise in Pathology – An Information Processing Study. J Am Med Inform Assoc 10(1):39-51, 2003.

[13] Fensel D, Benjamins V, Decker S, et al. The Component Model of UPML in a Nutshell. Proceedings of the First Working IFIP Conference on Software Architecture (WICSA1), San Antonio, Texas 1999, Kluwer.

[14] Cork RD, Detmer WM, and Friedman CP. Development and initial validation of an instrument to measure physicians' use of, knowledge about, and attitudes toward computers. J Am Med Inform Assoc. 5(2):164-76,1998.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Mining Data and Modelling Social Capital in Virtual Learning Communities

Ben K. DANIEL[1], Gordon I. McCALLA[1], Richard A. SCHWIER[2]
ARIES Research Laboratory[1]
Department of Computer Science, University of Saskatchewan
Educational Communication and Technology[2], University of Saskatchewan
3 Campus Drive, S7N 5A4, Saskatoon, Canada

Abstract. This paper describes the use of content analysis and Bayesian Belief Network (BBN) techniques aimed at modelling social capital (SC) in virtual learning communities (VLCs). An initial BBN model of online SC based on previous work is presented. Transcripts drawn from two VLCs were analysed and inferences were drawn to build scenarios to train and update the model. The paper presents three main contributions. First, it extends the understanding of SC to VLCs. Second; it offers a methodology for studying SC in VLCs. Third the paper presents a computational model of SC that can be used in the future to understand various social issues critical to effective interactions in VLCs.

1. Introduction

Social capital (SC) has recently emerged as an important interdisciplinary research area. SC is frequently used as a framework for understanding various social networking issues in physical communities and distributed groups. Researchers in the social sciences and humanities have used SC to understand trust, shared understanding, reciprocal relationships, social network structures, etc. Despite such research, little has been done to investigate SC in virtual learning communities (VLCs).

SC in VLCs can be defined as a web of positive or negative relationships within a group. Research into SC in physical communities shows that SC allows people to cooperate and resolve shared problems more easily [19]. Putnam [14] has pointed out that SC greases the wheel that allows communities to advance smoothly. Prusak and Cohen [13] have further suggested that when people preserve continuous interaction, they can sustain SC which can in turn enable them to develop trusting relationships. Further, in VLCs, SC can enable people to make connections with other individuals in other communities [14]. SC also helps individuals manage and filter relevant information and can enable people in a community to effectively communicate with each other and share knowledge [3].

This paper describes the use of content analysis and Bayesian Belief Network (BBN) techniques to develop a model of SC in VLCs. An initial BBN model for SC based on previous work [4] is presented. Transcripts of interaction drawn from two VLCs were used to train and validate the model. Changes in the model were observed and results are discussed.

2. Content Analysis

The goal of content analysis is to determine the presence of words, concepts, and patterns within a large body of text or sets of texts [17]. Content analysis involves the application of systematic and replicable techniques for compressing a large body of text into few categories based on explicit rules of coding [6] [16]. Researchers have used content analysis to understand data generated from interaction in computer-mediated collaborative learning environments [2] [15] [18]. Themes, sentences, paragraphs, messages, and propositions are normally used for categorizing texts and they are treated as the basic units of analysis [16]. In addition, the various units of analysis can serve as coding schemes enabling researchers to break down dialogues into meaningful concepts that can be further studied.

The variations in coding schemes and levels of analysis often create reliability and validity problems. Furthermore, content analysis approaches are generally cumbersome and labour intensive. However, a combination of content analysis and machine learning techniques can help to model dependency relationships and causal relationships among data.

2.1. Using Bayesian Belief Networks to Build Models

In artificial intelligence in education (AIED) models are used for diagnosing learners to enable the building of tools to support learning [9]. Models can also be used to represent various educational systems. Barker [1] summarized three uses of models within AIED: models as scientific tools for understanding learning problems; models as components of educational systems; and models as educational artefacts.

A Bayesian Belief Networks (BBN) is one of the techniques for building models. BBNs are directed acyclic graphs (DAGs) composed of nodes and directed arrows [12]. Nodes in BBNs represent random variables and the directed edges (arrows) between pairs of nodes indicate relationships among the variables. BBNs can be used for making qualitative inferences without the computational inefficiencies of traditional joint probability determinations [13]. Researchers have used BBN techniques for various purposes. For example BBNs have been used for student modelling [20] and user modelling [21]. We have begun to investigate how BBNs can model SC in virtual communities [4].

3. Modelling Social Capital in Virtual Learning Communities

The procedure for examining SC in VLC first involved synthesis of previous and current research on SC in physical communities, singling out the most important variables and establishing logical relationships among the variables. The main variables include: the type of community, attitudes, interaction, shared understanding, demographic cultural awareness, professional cultural awareness, task knowledge awareness, and individual capability awareness, norms, and trust. We represented various degrees of influence by the letters S (strong), M (medium), and W (weak). The signs + and - represent positive and negative relationships. The relationships among the variables were mapped into a BBN for SC (see figure 1).

Table 1. presents the key variables of SC and their definitions

Variable Name	**Variable Definition**	**Variable States**
Interaction	A mutual or reciprocal action between two or more agents determined by the number of messages sent and received	Present/Absent
Attitudes	Individuals' general perception about each other and others' actions	Positive/Negative
Community Type	The type of environment, tools, goals, and tasks that define the group	Virtual learning community (VLC) and Distributed community of practice (DCoP)
Shared Understanding	A mutual agreement/consensus between two or more agents about the meaning of an object	High/Low
Awareness	Knowledge of people, tasks, or environment and or all of the above	Present/Absent
Demographic Awareness	Knowledge of an individual: country of origin, language and location	Present/Absent
Professional Cultural Awareness	Knowledge of people’s background training, affiliation etc.	Present/Absent

Competence Awareness	Knowledge about an individual's capabilities, competencies, and skills	Present/Absent
Capability Awareness	Knowledge of people's competences and skills in regards to performing a particular task	Present/Absent
Social Protocols/Norms	The mutually agreed upon, acceptable and unacceptable ways of behaviour in a community	Present/Absent
Capability Awareness	Knowledge of people's competences and skills in regards to performing a particular task	Present/Absent
Social Protocols/Norms	The mutually agreed upon, acceptable and unacceptable ways of behaviour in a community	Present/Absent
Trust	A particular level of certainty or confidence with which an agent use to assess the action of another agent.	High/Low

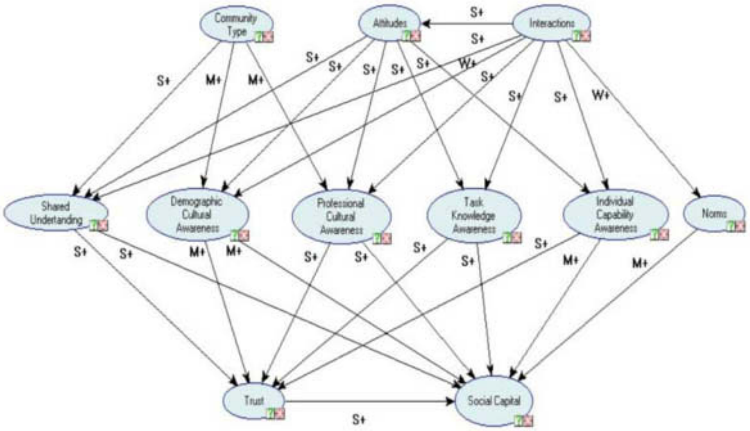


Figure 1. The Initial Model of Social Capital in Virtual Learning Communities [4]

3.1. Computing the Initial Probability Values

The probability values were obtained through adding weights to the values of the variables depending on the number of parents and the strength of the relationship between particular parents and children. For example, if there are positive relationships between two variables, the weights associated with each degree of influence are determined by establishing a threshold value associated with each degree of influence. The threshold values correspond to the highest probability value that a child could reach under a certain degree of influence from parents. For instance if Attitudes

Table 2. Threshold values and weights with two parents

Degree of influence	Thresholds	Weights
Strong	1-α = 1 - 0.02 = 0.98	(0.98-0.5) / 2 = 0.48 / 2 = 0.24
Medium	0.8	(0.8-0.5) / 2 =0.3 / 2 = 0.15
Weak	0.6	(0.6-0.5) / 2 =0.1 / 2 = 0.05

and Interactions have positive and strong (S+) relationships with Knowledge Awareness, the evidence of positive interactions and positive attitudes will produce a conditional probability value for Knowledge Awareness of 0.98 (threshold value for strong = 0.98).

The weights were obtained by subtracting a base value (1 / number of parents, 0.5 in this case) from the threshold value associated to the degree of influence and dividing the result by the number of parents (i.e. (0.98 - 0.5) / 2 = 0.48 / 2 = 0.24). Table 2 shows the threshold values and weights used in this example. Since it is more likely that a certain degree of uncertainty can exist, a value of α = 0.02 leaves some room for uncertainty when considering evidence coming from positive and strong relationships.

3.2. Testing the Bayesian Belief Network Model

In order to experiment with the model developed in [4], further scenarios were developed based on results obtained from studying two different virtual communities. One community, see you see me (CUCME), involved a group of individuals who regularly interacted on various issues using textual and visual technologies (video-cams). In the CUCME community there were no explicit goals but instead individuals were drawn together on a regular basis to interact socially. Themes that emerged from the analysis of the transcripts included economics, social issues, politics, food, religion, and technology etc. Table 3 shows the

Table 3. Frequency of messages observed in relation to each variable in the CUCME VC

Variable Name	Frequency	Percentage
Demographic Awareness	17	2.77
Economic	14	2.28
Food	12	1.96
Information Exchange	7	1.14
Social	45	7.35
Technology	7	1.14
Community Language	50	8.16
Hospitality	33	5.39
Use of Simile	21	3.43
Interaction	406	66.33
Total	**612**	**99.95**

number of messages in each category found in the transcripts, their percentage of the whole, and the mean.

The second community we studied consisted of graduate students learning theories and philosophies of educational technology. Unlike the first community, students in this community occasionally met face-to-face and they had explicit learning goals (complete course requirements) and protocols (set by the instructor of the course) of interactions. Bulletin boards and email were also used for interaction. The results of the analysis of the transcripts of this community's interactions were broken down into themes and are summarised in table 4.

Table 4. Frequency of messages observed in relation to each variable in the VLC

Variable Name	**Frequency**	**Percentage**
Interaction	100	9.12
Professional Awareness	15	1.36
Knowledge Awareness	8	0.72
Sociocultural Awareness	14	1.27
Technology	15	1.36
Hospitality	59	5.38
Shared Understanding	117	10.67
Information exchange	656	59.85
Social Protocols	112	10.21
Total	**1096**	**99.94**

4. Results and Discussion

The various themes that emerged from the analysis of the transcripts taken from interawere used to develop a number of scenarios which in turn were used to tweak the probability values in the model. A scenario refers to a written synopsis of inferences drawn from the results of the transcripts. A scenario was developed from the CUCME findings based on the following observations: high of interaction, high value of demographic awareness. The values of interaction, demographic awareness were tweaked in the initial model to reflect *positive state* and *present state* respectively. Our goal was to observe the level of shared understanding in the BBN model using the scenario described above.

After tweaking the variables based on the scenario, the model was updated. The results showed an increase in the posterior probability values of shared understanding i.e. P (shared understanding) = 0.915. And since shared understanding is also a parent of trust and SC, the probabilities of trust and SC have correspondingly increased P (trust) = 0.92 and P (SC) = 0.75. Similarly, evidence of negative interaction and negative attitudes in the CUCME community decreased the probabilities of P (shared understanding) = 0.639, P (trust) = 0.548 and P (SC) = 0.311. The results demonstrate dependency between the three variables.

In the second VLC (the graduate course) only five variables that were dominant in the BBN model (interaction, professional awareness, knowledge awareness, shared understanding and social protocols) were inferred from the results, and scenarios were developed around those variables. For example we want to examine the level of SC in a community with a high level of interaction (meaning that interaction is positive), and where individuals are exposed to each other well enough to know who knows what and works where, but are not aware of the demographic backgrounds of participants (various forms of awareness). Setting these variables in the model, we obtained, P (shared understanding) = 0.758, P (trust) = 0.899 and P (SC) = 0.717. The increase in the probabilities of shared understanding, trust and SC in this community given various kinds of awareness, but not demographic awareness, can be explained by the fact that this community has explicit learning goals, and that individuals are able to develop trusting relationship based on the information about what individuals know and are capable of doing rather than demographic information (where an individual is from etc.).

5. Conclusion

Using content analysis and BBN techniques, we have demonstrated how to model SC in VLCs. We have also shown how to update the model using scenarios that can be developed from the results obtained from natural interactions in virtual communities. Inferences from the posterior probabilities obtained from the scenaros suggest that within a specific type of virtual community, the level of SC can vary according to the level of shared understanding. Further, different forms of awareness seem to have different degrees of influence on SC. For example, in the CUCME demographic awareness seems to be an influential factor in the variation of SC. Moreover, in the graduate course VLC, where there are explicit goals and limited time to achieve those goals, members can be motivated to participate and engage in knowledge sharing activities and so demographic awareness can have a little influence on SC.

The Bayesian model presented in this paper adequately represented the scenarios developed from the results obtained from the two data sets. We are continuing to analyse interaction patterns in other VLCs, and will develop more scenarios to refine our model.

Acknowledgement

We would like to thank the Natural Sciences and Engineering Research Council of Canada (NSERCC) as well as the Social Sciences and Humanities Research Council of Canada (SSHRCC) for their financial support for this research.

References

[1] M. Baker (2000). The roles of models in Artificial Intelligence and Education research: A prospective view. *International Journal of Artificial Intelligence in Education* (11),123-143.

[2] B. Barros & F. Verdejo (2000). Analysing students interaction processes in order to improve collaboration. The DEGREE approach. *International Journal of artificial inteligence in education*, (11), pp. 221-241

[3] B.K. Daniel, R.A. Schwier & G. I. McCalla (2003). Social capital in virtual learning communities and distributed communities of practice. *Canadian Journal of Learning and Technology, 29(3*), 113-139.

[4] B.K. Daniel, D. J. Zapata-Rivera & G. I. McCalla (2003). A Bayesian computational model of social capital in virtual communities. In, M. Huysman, E.Wenger and V. Wulf Communities and Technologies, pp.287-305. London: Kluwer Publishers.

[5] Freeman, L. C. (2000), Visualizing social networks, Journal of Social Structure, Available: [http://zeeb.library.cmu.edu: 7850/JoSS/article.html]

[6] K. Krippendorf (1980). *Content analysis: An introduction to its methodology.* Beverly Hills, CA: Sage Publications.

[7] C. Lacave and F. J. Diez (2002). Explanation for causal Bayesian networks in Elvira. In Proceedings of the Workshop on Intelligent Data Analysis in Medicine and Pharmacology (IDAMAP-2002), Lyon, France.

[8] K. Laskey and S. Mahoney (1997). *Network Fragments: Representing Knowledge for Constructing Probabilistic Models*, Uncertainty in Artificial Intelligence: Proceedings of the Thirteenth Conference.

[9] G. I. McCalla (2000). The fragmentation of culture, learning, teaching and technology: Implications for artificial intelligence in education research. *International Journal of Artificial Intelligence in Education*, 11(2), 177-196.

[10] J. Nahapiet & S. Ghoshal (1998). Social capital, intellectual capital and the organizational advantage. *Academy of Management Review*, (23)(2) 242- 266.

[11] D. Niedermayer (1998) *An Introduction to Bayesian Networks and their Contemporary Applications*. Retrieved May, 6th 2004 from: : [http://www.niedermayer.ca/papers/bayesian/bayes.html]

[12] J. Pearl (1988) Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann Publishers, San Mateo, CA.

[13] L. Prusak & D. Cohen (2001). *In good company: How social capital makes organizations work.* Boston, MA: Harvard Business School Press.

[14] R. Putnam (2000). *Bowling alone: The collapse and revival of American community.* New York: Simon Schuster.

[15] A. Ravenscroft & R. Pilkington (2000). Investigation by design: developing dialogues models to support support reasoning and conceptual change. *International journal of artificial intellignece in education*, 11-273-298.

[16] L. Rourke T. Anderson, D.R. Garrison. and W. Archer (2001). Methodological issues in the analysis of computer conference transcripts. *International Journal of Artificial Intelligence in Education, (12) 8-22.*

[17] S. Stemler (2001). An overview of content analysis. *Practical Assessment, Research & Evaluation*, 7(17). Retrieved October 19, 2003 from [http://edresearch.org/pare/getvn.asp?v=7&n=17].

[18] A. Soller and A. Lesgold (2003). A computational approach to analyzing Online knowledge sharing interaction. *Proceedings of Artificial Intelligence in Education pp.* 253-260.*, Sydney, Australia.*

[19] World Bank (1999). *Social capital research group*. Retrieved May 29, 2003, from http://www.worldbank.org/poverty/scapital/.

[20] J.D. Zapata-Rivera (2002) cbCPT: Knowledge Engineering Support for CPTs in Bayesian Networks. Canadian Conference on AI 2002: 368-370

[21] I. Zukerman, D. W., Albreacht and A.E. Nelson (1999). Predictiing users' requests on the WWW. In UM 1999, Proceedings of the 7th international conference on user modelling, Banf, Canada, pp, 275-284.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Tradeoff analysis between knowledge assessment approaches

Michel C. Desmarais, Shunkai Fu, Xiaoming Pu
École Polytechnique de Montréal

Abstract. The problem of modeling and assessing an individual's ability level is central to learning environments. Numerous approaches exists to this end. Computer Adaptive Testing (CAT) techniques, such as IRT and Bayesian posterior updating, are amongst the early approaches. Bayesian networks and graphs models are more recent approaches to this problem. These frameworks differ on their expressiveness and on their ability to automate model building and calibration with empirical data. We discuss the implication of expressiveness and data-driven properties of different frameworks, and analyze how it affects the applicability and accuracy of the knowledge assessment process. We conjecture that although expressive models such as Bayesian networks provide better cognitive diagnostic ability, their applicability, reliability, and accuracy is strongly affected by the knowledge engineering effort they require. We conclude with a comparative analysis of data driven approaches and provide empirical estimates of their respective performance for two data sets.

Keywords. Student models, Bayesian inference, graphical models, adaptive testing, CAT, IRT, Bayesian networks

1. Introduction

Assessing the user's mastery level with respect to one or more abilities is a key issue in learning environments. Any system that aims to provide intelligent help/assistance to a user is bound to model what that person already knows and doesn't know.

The Item Response Theory (IRT) emerged as one of the earliest and most successful approaches to perform such assessment [2]. The field of Computer Adaptive Testing, which aims to assess an individual's mastery of a subject domain with the least number of question items administered, has relied on this theory since its conception.

IRT has the characteristic of being data driven: knowledge assessment is purely based on model calibration with sample data. Model building is limited to defining which item belongs to which skill dimension. These are important characteristics that IRT shares with other student modeling approaches such as Bayesian posterior updates [17] and POKS [5]. We return to this issue later.

Curiously, until fairly recently, the field of intelligent learning environments did not adopt the IRT approach to modeling the learner's expertise, even though this approach was cognitively and mathematically sound. Instead, techniques known as "overlay models" [3] and "stereotypes" [16] were used to model what the user knows. It remains speculative to explain why the research community working on intelligent learning applications has, at least initially, largely ignored the work on IRT and other data driven approaches, but we can evoke some possibilities:

- training data that could prove difficult to collect if large samples are required;
- IRT requires numerical methods (eg. multi-parameters maximum likelihood estimation) that were non trivial to implement and not widely available as software packages until recently;
- the AI community was not familiar with the field from which IRT comes from, namely psychometric research;
- intelligent learning applications focused on fine grained mastery of specific concepts and student misconceptions in order to allow highly specific help/tutoring content to be delivered; IRT was not designed for such fine grained assessment but focuses instead on the determining the mastery of one, or a few, ability dimensions.

However, in the last decade, this situation has changed considerably. Overlay and stereotype-based models are no longer the standard for performing knowledge assessments in AI-based learning systems. Approaches that better manage the uncertainty inherent to student assessment, such as probabilistic graphical models and

Bayesian networks, are now favored. In fact, researchers from the psychometric and the Student/User Modeling communities are recently working on common approaches. These approaches rely on probabilistic graph models that share many commonalities with IRT-based models, or encompass and extend such models [1,11,15,12]. Reflecting on these last developments, we can envision that the data driven and the probabilistic/statistical models, of which IRT is an early example, and the fine grained diagnostic approaches, typical of Intelligent Learning Environments, are gradually merging. In doing so, they can yield powerful models and raise the hope of combining the best of both fields, namely cognitively and mathematically sound approaches that are amenable to statistical parameter estimation (i.e. full automation), and high modeling flexibility necessary for intelligent learning environments.

We review some of the emerging models and compare their respective advantages from a qualitative perspective, and conclude with a performance analysis of three data driven approaches over two domains of assessment.

2. Qualitative Factors

Student modeling approaches differ over a number of dimensions that can determine the choice of a specific technique in a given context of application. These dimensions are summarized below:

Flexibility and expressiveness: As hinted above, AI-based systems often rely on fine-grained assessment of abilities and misconceptions. Although global skill dimensions are appropriate in the context of assessing general mastery of a subject matter, many learning environments will require more fine-grained assessment.

Cost of model definition: Fine-grained models such as those found in Bayesian Networks (see, for example, Vomlel [18] and Conati [4]) require considerable expert modeling effort. On the contrary, data driven approaches such as IRT can completely waive the knowledge engineering effort. Because of the modeling effort, fine-grained models can prove overly costly for many applications.

Scalability: The number of concepts/skills and test items that can be modeled in a single system is another factor that weights into evaluating the appropriateness of an approach. The underlying model in IRT allows good scalability to large tests and for a limited number of ability dimensions. For fine grained student models, this factor is more difficult to assess and must be addressed on a per case basis. For example, in a Bayesian Network where items and concepts are highly interconnected, complexity grows rapidly and can be a significant obstacle to scalability.

Cost of updating: The business of skill assessment is often confronted with frequent updating to avoid over exposure of the same test items. Moreover, in domains where the skills evolve rapidly, such as in technical training, new items and concepts must be introduced regularly. Approaches that reduce the cost of updating the models are at significant advantage here. This issue is closely tied to the knowledge engineering effort required and the ability of the model to be constructed and parametrized with a small data sample.

Accuracy of prediction: Student modeling applications such as Computer Adaptive Testing (CAT) are critically dependent on the ability of the model to provide an accurate assessment with the least number of questions. Models that can yield confidence intervals, or the degree of uncertainty of their inferences/assessment, are thus very important in this field as well as in many context in which measures of accuracy is relevant.

Reliability and sensitivity to external factors: A factor that is often difficult to assess and overlooked is the reliability of a model to environmental factors such as the skills of the knowledge engineer, the robustness to noise in the model, and to noise in the data used to calibrate a model. Extensive research in IRT has been conducted on the issue of reliability and robustness under different conditions, but little has been done in intelligent learning environments.

Mathematical foundations: The advantages of formal and mathematical models need not be defended. Models that rely on sound and rigorous mathematical foundations are generally considered better candidates over *ad hoc* models without such qualities because they provide better support to assess accuracy and reliability, and they can often be automated using standard numerical modeling techniques and software packages. Both the Bayesian Network and IRT approaches do fairly well on this ground, but they also make a number of assumptions that can temper their applicability.

Approximations, assumptions, and hypothesis: In the complex field of cognitive and skill modeling, all models must make a number of simplifying assumptions, hypothesis, or approximations in order to be applicable. This is also true of Bayesian modeling in general. Of course, the more assumptions and approximations are made, the less accurate and reliable a model becomes. This issue is closely linked to the reliability and sensitivity one. Some approach may work well in one context and poorly in another because of violated assumptions.

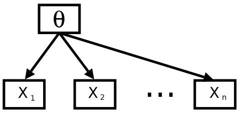


Figure 1. Graphical representation of the links between θ, the examinee's mastery or ability level, and $\{X_1, X_2, ..., X_n\}$, the test items.

These factors will determine the value of a student modeling approach. A modeling approach that requires highly skilled experts in Bayesian modeling, combined with expert content knowledge, and that performs poorly if some modeling errors are introduced, will be much less appealing than an approach that can be fully automated using small samples to build and calibrate the model, whose reliably is good and measurable, and yet, that permits fine grained cognitive modeling.

3. Qualitative Comparison of Approaches

The previous section establishes the qualitative factors by which we compare different approaches to student skill modeling. This section pursues with an analysis of how models fare with respect to the factors mentioned. A more specific quantitative comparison will follow.

The student models we focus on are (1) IRT, (2) a simple Bayesian posterior probability update, (3) a graphical model that links items among themselves and uses a Bayesian update algorithm (POKS), and (4) more complex Bayesian and graphical models that link together concept and misconceptions (hidden variables), and items (evidence nodes) within the same structure.

3.1. Bayesian Posterior Updates

The simplest approach to assessing mastery of a subject matter is the Bayesian posterior update. It consists in the application of Bayes rule to determine the posterior probability: $P(m|X_1, X_2, \ldots, X_n)$, where m stands for *master* and $X_1, X_2, \ldots, X_n$ is the response sequence after n item responses are given. According to Bayes theorem and under strong independence assumptions, the posterior probability of m given the observation of item X_i is determined by:

$$P(m|X_i) = \frac{P(X_i|m)\ P(m)}{P(X_i|m)\ P(m) + P(X_i|\neg m)\ (1 - P(m))} \qquad (1)$$

$P(m|X_i)$ will serve as the next value for $P(m)$ for computing $P(m|X_{i+1})$. The initial and conditional probabilities, $P(m)$ and $P(m|X_i)$, are obtained from sample data. We refer the reader to Rudner [17] for further details.

The approach can be graphically represented by figure 1 and by considering θ as the *mastery* node and $\{X_1, X_2, ..., X_n\}$ as the test items. The interpretation of this graph is that θ, the event that the student masters the subject matter, will influence the probability of correctly answering each test items. Almond [1] shows that this graph also corresponds to the IRT model, although the probability updating scheme is different. More on this later in section 3.2.

This approach has many advantages that stem from its simplicity. It does not require knowledge engineering and can be fully automated and calibrated with small data sets. It is also computationally and conceptually very simple.

That simplicity comes at the price of low granularity and strong assumptions. In equation (1), the student model is limited two states, *master* or *non-master* with regards to a subject matter[1]. The model also makes the assumption that all test items have similar discrimination power, whereas it is common to find items significantly more discriminant than others.

Although figure 1 illustrates a single dimension example, multiple dimensions, or multiple concepts, can readily be modeled with this approach. Each concept or subject matter, s, can be represented by their respective θ_s. Moreover, the model can be extended to more than two states, although a larger data set will be necessary to obtain equivalent accuracy as in a two-states model. Some intelligent tutoring systems have used such extensions to the basic principle of Bayesian posterior probability updates to build intelligent learning environments [9,1]. Some also relied on subjective assessments to derive the conditional probabilities, but that strategy is highly subject to human biases and low agreement amongst experts that can result in poor accuracy and low reliability.

[1] Mastery is determined by an arbitrary passing score.

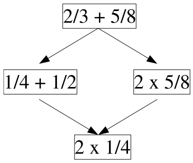


Figure 2. Graphical example of the interrelationships between abilities to solve different arithmetic problems. The graph represents the order in which items are mastered.

3.2. *Item Response Theory*

IRT can be considered as a graphical network similar to the one in figure 1. However, in contrast to the Bayesian posterior update method, the variable θ represents an ability level on a continuous scale. The probability of succeeding an item X_i is determined by a logistic function named the Item Characteristic Curve (ICC)[2]:

$$P(z_i \mid \theta) = \frac{1}{1 + e^{-a_i(\theta - b_i)}} \tag{2}$$

Note that this particular function is called the "two-parameter logistic model". Other variants exists, dropping parameters a and b, or adding a guessing parameter c. The function defines an 'S' shaped curve where the probability $P(X_i)$ increases as a function of θ, as one would expect. The parameter a determines the slope of increase around a value of θ determined by the second parameter, b.

The value of θ is determined by maximizing the likelihood of the responses provided by the student, generally using a maximum-likelihood numerical method. IRT is a well documented and details can be found in Reckase [14].

IRT has the advantage of being a fully automated method that can be calibrated with relatively small data set, depending on the desired accuracy of the assessment. Contrary to the Bayesian posterior update approach in section 3.1, the two-parameter IRT model takes into account the discrimination factor of individual test items, and it models ability on a continuous scale as opposed to a dichomotous variable, or a multinomial variable when the model is extended. This last property of the model also means that a greater accuracy can be expected for computing $P(X_i|\theta)$. That information can, in turn, be useful for the purpose of computing the most informative test items or adjusting item difficulty. Finally and as mentioned, the model can be extended for multidimensionality. In short, it is a more sophisticated model than the Bayesian posterior updating model, but it does not allow fine-grained modeling of a large number of dimensions such as found in some intelligent tutoring systems where individual concepts and misconceptions are often modeled.

3.3. *Probabilistic Graphs Models*

Figure 1's graph model is limited to a single ability dimension and test items are singly-connected the ability node. However, graph models can also embed specific concepts and misconceptions in addition to general skill dimension and test items. The network structure can be a multilevel tree structure. Test items can be connected together in a directed graph such as figure 2's structure. We refer to such extensions as probabilistic graph models (for a more detailed discussion on the subject, see Almond [1]).

To model fine-grained skill acquisition, such as individual concepts and misconceptions, probabilistic graphical models are arguably the preferred approach nowadays. Such models represent the domain of skills/misconceptions as the set of nodes in the graph, $\{X_1, X_2, ..., X_n\}$. A student model consists in assigning a probability to each of the node's value. The arcs of the graph represent the interrelationships amongst these nodes. The semantics of the arcs varies according to the approach, but it necessarily has the effect that changes occurring in the probability of a node affects neighboring nodes and, under some conditions according to the inference approach, it can propagate further.

3.4. *Item to Item Graph Models*

One probabilistic graph model approach is to link test items among themselves. The domain of expertise is thus defined solely by observable nodes. A "latent" ability (i.e. non directly observable) can be defined by a subset of nodes, possibly carrying different weights. This is essentially the principle behind exam tests where questions can carry weights and where the mastery of a topic in the exam is defined as the weighted sum of success items.

[2] The ICC curve can also be defined after what is known as the normal ogive model but the logistic function is nowadays preferred for its greater computational tractability.

Item to item graph models derive probabilities of mastery of an item given the partial order structure of items (as in figure 2), and given the items observed so far. The semantics of links in such structures simply represents the order in which items are mastered. The cognitive basis behind such an approach is the Knowledge space theory [7], which states that the order in which people learn to master knowledge items is constrained by an AND/OR graph structure. The example in figure 2 illustrates the order in which we could expect people to learn to solve these simple arithmetic problems. For example, we learn to solve $2 \times 1/4$ before we can solve $1/4+1/2$, but the order is not clearly defined between abilities for solving $2 \times 5/8$ and solving $1/4 + 1/2$. Figure 2 is, in fact, a directed acyclic graph (DAG), not an AND/OR graph, but it does capture the partial ordering of mastery amongst knowledge items and allows to make valuable inferences. What it does not capture, are alternative methods of mastery. We refer the reader to Falmagne et al. [7] for more details on this theory.

Some researchers adopted this type of graph representation to perform knowledge assessment. Kambouri et al. [10] used a combination of data driven and knowledge engineering approach to build knowledge structures (AND/OR graphs), whereas Desmarais et al. [5] used a data driven only, automated approach to building a simplified (AND graph instead of AND/OR graph) version of knowledge structures represented as DAG. That approach is named Partial Order Knowledge Structures (POKS). We will compare the POKS performance to the IRT and the Bayesian posterior update approaches in section 4.

The advantage of leaving out the latent abilities from the graph structure is that model construction can be fully automated (at least for the POKS approach). It also involves benefits in terms of reliability and replicability by avoiding expert-based model building and subjective and individual biases. The disadvantages is the loss of explicit links between concepts or misconceptions in the graph structure. However, latent abilities (concepts) can later be derived by determining the items that are evidence for given concepts. For example, if concept C_1 has evidence nodes X_1, X_2, X_3, mastery of C_1 can be defined as a weighted sum of the probabilities of its evidence nodes: $C_1 = w_1X_1 + w_2X_2 + w_3X_3$.

We can argue that the reintroducion of latent abilities (concepts) incurs a knowledge engineering effort that we claimed is initially waived by the item to item approach, and thus that we simply hereby delay the knowledge engineering effort. Although it is true that there is a knowledge engineering effort that cannot be avoided when introducing concepts, namely linking concepts to some means of assessing them (cf. test items), there are significant differences. First, defining mastery of a concept as a weighted sum of items is much simpler than building a Bayesian model betwen items and concepts. To a certain extent, it is a process that teachers frequently go through when constructing an exam that covers different topics. On the contrary, estimating joint conditional probabilities between multiple items and multiple concepts is a much more difficult task. Subjective estimates of such joint conditional probabilities is unreliable and subject to biases. Yet, estimating those probabilities from data is also difficult because we do not observe mastery concepts directly. They have to be treated as latent variables which significantly complexifies their modeling. Bayesian modeling with latent variables is limited to expert, contrary to defining concepts as a weighted sum of test items.

3.5. Concept and Misconception Nodes Graph Models

Graph structures that also include concept and misconceptions nodes in addition to test items can derive the probability of success in a more sophisticated manner than the item to item graph models described above. Probability of mastery of a concept can be determined by estimated mastery of other concepts and by the presence of misconceptions in the student model. Most research in intelligent learning environments used different variations of this general approach to build graph models and Bayesian networks to perform student expertise assessment (to list only a few: [18,4,13,11]).

By modeling the interdependencies between concepts of different level of granularity and abstractions, misconceptions, and test items that represent evidence, it comes as no surprise that a wide variety of modeling approaches are introduced. We will not attempt to further categorize graph models and Bayesian networks here, but try to summarize some general observations that can be made on these.

A first observation is that the student models can comprise fine-grained and highly relevant pedagogical information such as misconceptions. It entails that detailed help or tutoring content can be delivered to the user once the student cognitive assessment is derived.

We also note that many approaches rely on a combination of data driven and knowledge engineering to derive the domain model. However, we know of no example that is completely data driven. This is understandable since detailed modeling of concepts and misconceptions necessarily requires pedagogical expertise. What can be data driven is the calibration of the model, namely the conditional probabilities in Bayesian networks.

The variety of approaches in using Bayesian networks and graph models to build student models that include concepts and misconceptions is much too large for a proper coverage in the space allotted here. Let us only conclude this section by mentioning that, although these approaches are currently more complex to build and to use, they have strong potential because of their flexibility. The effort required is most appropriate for knowledge domains that are stable such as mathematics teaching.

4. Performance Comparison

In the previous sections, we attempt to draw a comparative picture of some student modeling approaches over dimensions such as data-driven vs human engineered models, which in turn has impacts on how appropriate is an approach for a given context. Very simple approaches based on Bayesian posterior updates, and slightly more sophisticated ones such as IRT and item to item graph structures, can be entirely data driven and require no knowledge engineering effort. By contrast, more complex structures involving concepts and misconceptions are not currently easily amenable to fully automated model construction, although model calibration is feasible in some cases.

We conducted an empirical comparison of the data driven approaches over two knowledge domains, the Unix shell commands and the French language. The approaches are briefly described and the results reported. First, a short description of the simulation method for the performance comparison is described.

4.1. Simulation Method

The performance comparison is based on the simulation of the question answering process. For each examinee, we simulate the adaptive questioning process with the examinees' actual responses[3]. The same process is repeated for every approach. After each item administered, we evaluate the accuracy of the examinee's classification as a *master* or *non master* according to a pre-defined passing score, for eg. 60%.

The choice of the next question is adaptive. Each approach uses a different method for determining the next question because the optimal method depends on the approach. We use the method for choosing the next question that yields the best result for each approach.

The performance score of each approach corresponds to the number of correctly classified examinees after i items are administered.

The simulations are made on two sets of data: (1) a 34 items test on the knowledge of Unix shell commands administered to 48 examinees, and (2) a 160 items test on French language administered to 41 examinees.

4.2. Bayesian Posterior Updates, IRT, and POKS Comparison

All approaches compared are well documented elsewhere and we limit their descriptions to brief overviews.

4.2.1. Bayesian Posterior Updates

The Bayesian posterior updates procedure consists in applying Bayes rules according to equation (1).

The choice of the next question to ask is the maximum discrimination measure [17]:

$$M_i = |\log(P(z_i|m)/P(z_i|\neg m))|$$

4.2.2. Item Response Theory (IRT)

The simulation uses the two-parameters logistic model version of IRT which corresponds to equation (2). Values for parameters a and b are calibrated using the sample data sets. Estimation of θ is performed with a maximum likelihood estimation procedure after each item is administered.

Choice of the next question corresponds is based on the Fisher information measure, which is the most widely used for the IRT approach and it was introduced early on in IRT [2]. The Fisher information is a function of θ and the parameters a and b of equation (2).

4.2.3. Partial Order Knowledge Structure (POKS)

The POKS method is described in Desmarais et al. [5] (see also [6]). It consists in inferring structures such as the one in figure 2 from the data sample. Updating of the conditional probabilities is based on Bayesian posterior probabilities of the parent nodes. Evidence is propagated from observed nodes (items answered) in accordance to Bayes rule for the nodes directly connected with the observed one. Evidence is further propagated to indirectly linked nodes according to an interpolation scheme known as the PROSPECTOR algorithm (see Giarratano [8]). For linear structures (eg. $A \rightarrow B \rightarrow C$), that approximation yields probability values equivalent to the direct application of Bayes rule. For other structures, the values will differ according to how valid are the assumptions of conditional independence of the POKS framework for the data set, and how accurate is the approximation. To a large extent, an empirical answer to this question is provided by the performance evaluation.

The choice of the next question is determined by the minimal entropy measure. The item chosen corresponds to the one that is expected to reduces the most the entropy of the test items set. The entropy measure is based on

[3] Taking care of removing from the calibration data the simulation's current examinee's data case in order to avoid over-calibration.

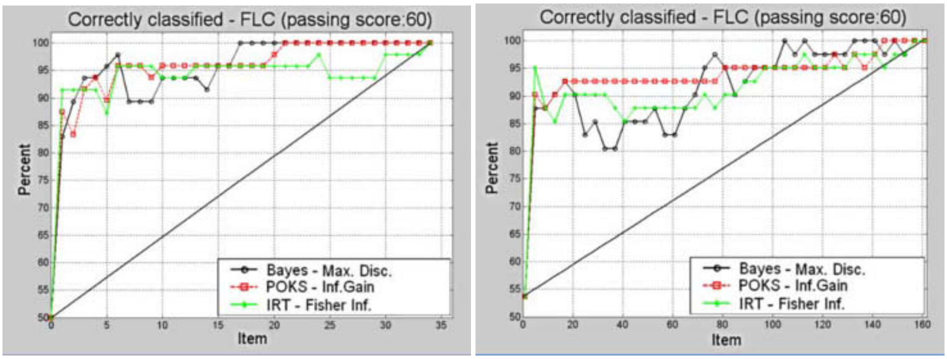


Figure 3. Performance comparison of three knowledge assessment methods.

the standard formula $-[p\log(p) + (1-p)\log(1-p)]$ and the test's entropy is the summation over all test item entropies. Test entropy value is highest if all items have a probability of 0.5 (i.e. maximum uncertainty), and it is 0 if all items have a probability of 1 or 0.

4.3. Results

The performance of the three approaches are compared in figure 3. It reports the results of the simulation for the Unix and French language tests comprised respectively of 34 and 160 items. The percentage of correctly classified examinees, averaged over 48 simulation cases for the Unix test and 41 for the French language one, are plotted as a function of the number of item responses. Passing score is 60%. The diagonal line is for baseline comparison.

Both plots start at 0 questions, which corresponds to the number of correctly classified examinees that correctly fall into the most likely state (master or non master) according to the sample. For the Unix test about half were master, thus the starting score is around 50%, whereas for the French test a little more than half were master. The x-axis end at the number of questions in the test and at a 100% correctly classified score, when all items are answered. After about 5 question items, all three approaches correctly classify more than 85% of examinees for both tests but, for the French test and after about 5 items, the POKS approach perform a little better than the Bayes posterior update and the IRT approaches. The Bayes approach also appears to be less reliable as the curve fluctuates more than the other two throughout the simulation.

Other simulations shows that POKS and IRT are in general better than Bayes posterior update at cutting scores varying from 50% to 70%[4], and that POKS is slightly better than IRT but not systematically (further details can be found in Desmarais et al. [6]).

5. Conclusion

Student models are gradually converging towards a probabilistic representation of mastery of skill sets. Automated and data driven models such as Bayesian posterior update, IRT, and Partial Order Knowledge Structures (POKS), limit their representation to observable test items. Subsets of items can be used to define higher level skills, but knowledge assessment is not based on concepts/skills directly. These approaches have the advantages of avoiding the knowledge engineering effort to building the student model. With this come further advantages such as avoidance of human biases and individual differences in modeling, the possibility of full automation and reduced costs for building and updating the models, and a reliability and accuracy that can better be measured and controlled as a function of sample size.

We show that the accuracy of the three data driven approaches for classifying examinees is relatively good. Even the simplest method, namely the Bayesian posterior updates, performs relatively well with small data samples below 50 cases, but it is less accurate and reliable than the other two.

Graphical models and Bayesian networks that include concept and misconception nodes provide more flexibility and diagnostic power than the data-driven approaches reviewed. However, they generally require a knowledge engineering effort that hampers their applicability and can also affect their accuracy. It would be interesting to have a Bayesian network approach to add to the comparison study to better assess their comparative accuracy. This paper aims to nurture some effort in this direction.

[4] Simulations beyond the 50% to 70% range is unreliable because almost all examinees are already correctly classified before any item is answered.

Acknowledgements

This work was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] Russell G. Almond and Robert J. Mislevy. Graphical models and computerized adaptive testing. *Applied Psychological Measurement*, 23(3):223–237, 1999.

[2] A. Birnbaum. Some latent trait models and their use in infering an examinee's ability. In F.M. Lord and M.R. Novick, editors, *Statistical Theories of Mental Test Scores*, pages 397–472. Addison-Wesley, Reading, MA, 1968.

[3] B. Carr and I. Goldstein. Overlays: A theory of modelling for computer aided instruction. Technical report, 1977.

[4] C. Conati, A. Gertner, and K. VanLehn. Using bayesian networks to manage uncertainty in student modeling. *User Modeling and User-Adapted Interaction*, 12(4):371–417, 2002.

[5] Michel C. Desmarais, Ameen Maluf, and Jiming Liu. User-expertise modeling with empirically derived probabilistic implication networks. *User Modeling and User-Adapted Interaction*, 5(3-4):283–315, 1995.

[6] Michel C. Desmarais and Xiaoming Pu. Computer adaptive testing: Comparison of a probabilistic network approach with item response theory. In *Proceedings of the 10th International Conference on User Modeling (UM'2005)*, page (to appear), Edinburg, July 24–30 2005.

[7] J.-C. Falmagne, M. Koppen, M. Villano, J.-P. Doignon, and L. Johannesen. Introduction to knowledge spaces: How to build test and search them. *Psychological Review*, 97:201–224, 1990.

[8] J.C. Giarratano and G. Riley. *Expert Systems: Principles and Programming (3rd edition).* PWS-KENT Publishing, Boston, MA, 1998.

[9] Anthony Jameson. Numerical uncertainty management in user and student modeling: An overview of systems and issues. *User Modeling and User-Adapted Interaction*, 5(3-4):193–251, 1995.

[10] Maria Kambouri, Mathieu Koppen, Michael Villano, and Jean-Claude Falmagne. Knowledge assessment: tapping human expertise by the query routine. *International Journal of Human-Computer Studies*, 40(1):119–151, 1994.

[11] Joel Martin and Kurt Vanlehn. Student assessment using bayesian nets. *International Journal of Human-Computer Studies*, 42(6):575–591, June 1995.

[12] Michael Mayo and Antonija Mitrovic. Optimising ITS behaviour with bayesian networks and decision theory. *International Journal of Artificial Intelligence in Education*, 12:124–153, 2001.

[13] Eva Millán and José Luis Pérez-de-la-Cruz. A bayesian diagnostic algorithm for student modeling and its evaluation. *User Modeling and User-Adapted Interaction*, 12(2–3):281–330,, 2002.

[14] M. D. Reckase. A linear logistic multidimensional model. In W. J. van der Linden and R. K. Hambleton, editors, *Handbook of modern item response theory*, pages 271–286. New York: Springer-Verlag, 1997.

[15] Jim Reye. Student modelling based on belief networks. *International Journal of Artificial Intelligence in Education*, 14:63–96, 2004.

[16] Elaine Rich. User modeling via stereotypes. *Cognitive Science*, 3:329–354, 1979.

[17] Lawrence M. Rudner. An examination of decision-theory adaptive testing procedures. In *Proceedings of American Educational Research Association*, pages 437–446, New Orleans, 1–5 2002.

[18] Jiří Vomlel. Bayesian networks in educational testing. *International Journal of Uncertainty, Fuzziness and Knowledge Based Systems*, 12(Supplementary Issue 1):83–100, 2004.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Natural Language Generation for Intelligent Tutoring Systems: a case study

Barbara Di Eugenio [a,1] Davide Fossati [a] Dan Yu [a] Susan Haller [b] Michael Glass [c]
[a] *University of Illinois, Chicago, IL, USA*
[b] *University of Wisconsin Parkside, Kenosha, WI, USA*
[c] *Valparaiso University,Valparaiso, IN, USA*

Abstract. To investigate whether Natural Language feedback improves learning, we developed two different feedback generation engines, that we systematically evaluated in a three way comparison that included the original system as well. We found that the system which intuitively produces the best language does engender the most learning. Specifically, it appears that presenting feedback at a more abstract level is responsible for the improvement.

Keywords. Intelligent Tutoring Systems. Feedback Generation.

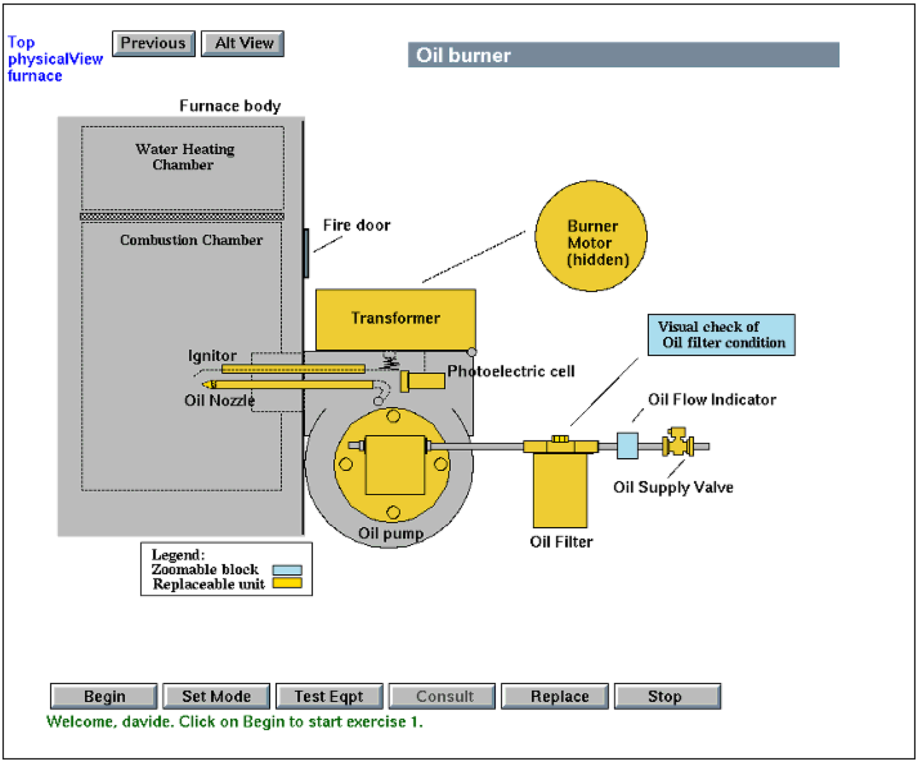
1. Introduction

The next generation of Intelligent Tutoring Systems (ITSs) will be able to engage the student in a fluent Natural Language (NL) dialogue. Many researchers are working in that direction [4,6,10,12,14]. However, it is an open question whether the NL interaction between students and an ITS does in fact improve learning, and if yes, what specific features of the NL interaction are responsible for the improvement. From an application point of view, it makes sense to focus on the most effective features of language, since deploying full-fledged dialogue interfaces is complex and costly.

Our work is among the first to show that a NL interaction improves learning. We added Natural Language Generation (NLG) capabilities to an existing ITS. We developed two different feedback generation engines, that we systematically evaluated in a three way comparison that included the original system as well. We focused on aggregation, i.e., on how lengthy information can be grouped and presented as more manageable chunks. We found that syntactic aggregation does not improve learning, but that functional aggregation, i.e. abstraction, does.

We will first discuss DIAG, the ITS shell we are using, and the two NLG systems we developed, *DIAG-NLP1*and *DIAG-NLP2*. Since the latter is based on a corpus study, we will briefly describe that as well. We will then discuss the formal evaluation we conducted and our results.

[1] Correspondence to: B. Di Eugenio, Computer Science (M/C 152), University of Illinois, 851 S. Morgan St., Chicago, IL, 60607, USA. Email: bdieugen@cs.uic.edu.



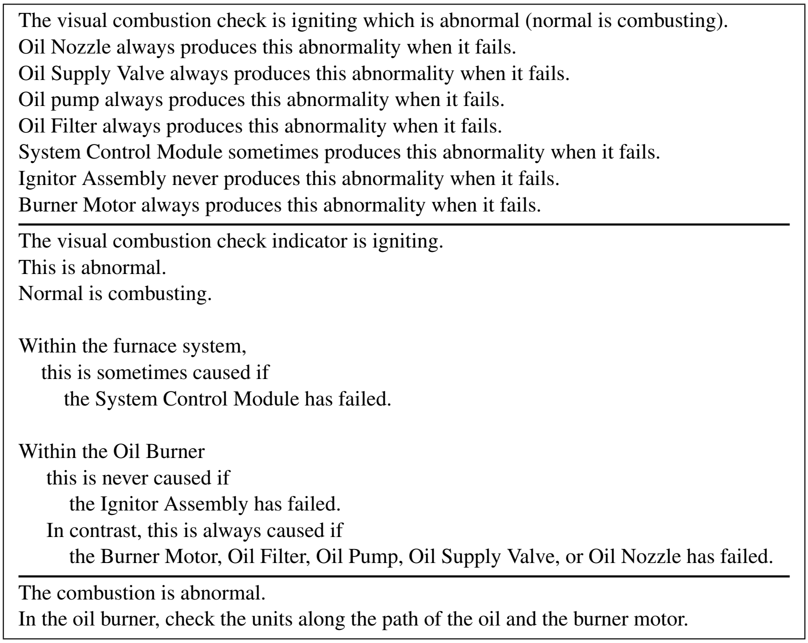
Welcome, davide. Click on Begin to start exercise 1.

Figure 1. The oil burner

2. Natural Language Generation for DIAG

DIAG [16] is a shell to build ITSs based on interactive graphical models that teach students to troubleshoot complex systems such as home heating and circuitry. DIAG integrates a functional model of the target system and qualitative information about the relations between symptoms and faulty parts (RUs) — RU stands for *replaceable unit*, because the only course of action for a student to fix the problem is to replace RUs in the graphical simulation. A DIAG application presents a student with a series of troubleshooting problems of increasing difficulty. The student tests *indicators* and tries to infer which RU may cause the abnormal states detected via the indicator readings. DIAG's educational philosphy is to push the student to select the most informative tests, and not to provide too much explicit information when asked for hints.

Fig. 1 shows the oil burner, one subsystem of the home heating system in *DIAG-orig*, our DIAG application. Fig. 1 includes indicators such as *Oil Flow indicator*, and many RUs such as *Oil Filter, Ignitor* etc. At any point, the student can consult the tutor via the Consult menu (cf. the Consult button in Fig. 1). There are two main types of queries: *ConsultInd(icator)* and *ConsultRU*. *ConsultInd* queries are used mainly when an indicator shows an abnormal reading, to obtain a hint regarding which RUs may cause the problem. DIAG discusses the RUs that should be most suspected given the symptoms the student has already observed. *ConsultRU* queries are mainly used to obtain feedback on the diagnosis that a certain RU is faulty. DIAG responds with an assessment of that diagnosis and provides evidence for it in terms of the symptoms that have been observed relative to that RU.



```
The visual combustion check is igniting which is abnormal (normal is combusting).
Oil Nozzle always produces this abnormality when it fails.
Oil Supply Valve always produces this abnormality when it fails.
Oil pump always produces this abnormality when it fails.
Oil Filter always produces this abnormality when it fails.
System Control Module sometimes produces this abnormality when it fails.
Ignitor Assembly never produces this abnormality when it fails.
Burner Motor always produces this abnormality when it fails.
```

```
The visual combustion check indicator is igniting.
This is abnormal.
Normal is combusting.

Within the furnace system,
   this is sometimes caused if
      the System Control Module has failed.

Within the Oil Burner
   this is never caused if
      the Ignitor Assembly has failed.
   In contrast, this is always caused if
      the Burner Motor, Oil Filter, Oil Pump, Oil Supply Valve, or Oil Nozzle has failed.
```

```
The combustion is abnormal.
In the oil burner, check the units along the path of the oil and the burner motor.
```

Figure 2. Answers to *ConsultInd* by *DIAG-orig*, *DIAG-NLP1*and *DIAG-NLP2*

DIAG uses very simple templates to assemble the text to present to the student. As a result, its feedback is highly repetitive and calls for improvements based on NLG techniques. The top parts of Figs. 2 and 3 show the replies provided by *DIAG-orig* to a *ConsultInd* on the *Visual Combustion Check*, and to a *ConsultRu* on the *Water Pump*.

Our goal in developing *DIAG-NLP1* and *DIAG-NLP2* was to assess whether simple, rapidly deployable NLG techniques would lead to measurable improvements in the student's learning. The only way we altered the interaction between student and system is the actual language that is presented in the output window. DIAG provides to *DIAG-NLP1* and *DIAG-NLP2* a file which contains the facts to be communicated – a *fact* is the basic unit of information that underlies each of the clauses in a reply by *DIAG-orig*. Both *DIAG-NLP1* and *DIAG-NLP2* use EXEMPLARS [17], an object-oriented, rule-based generator. EXEMPLARS rules are meant to capture an exemplary way of achieving a communicative goal in a given context.

DIAG-NLP1, which is fully described in [7], (i) introduces syntactic aggregation – i.e., uses syntactic means, such as plurals and ellipsis, to group information [13,15] – and what we call *structural* aggregation, i.e., groups parts according to the structure of the system; (ii) generates some referring expressions; (iii) models a few rhetorical relations (e.g. *in contrast* in Fig. 2); and (iv) improves the format of the output.

The middle part of Fig. 2 shows the output produced by *DIAG-NLP1*(omitted in Fig. 3 because of space constraints). The RUs of interest are grouped by the system modules that contain them (Oil Burner and Furnace System), and by the likelihood that a certain RU causes the observed symptoms. The revised answer highlights that the *Ignitor Assembly* cannot cause the symptom.

Water pump is a very poor suspect.
Some symptoms you have seen conflict with that theory.
Water pump sound was normal.
This normal indication never results when this unit fails.
Visual combustion check was igniting.
This abnormal indication never results when this unit fails.
Burner Motor RMP Gauge was 525.
This normal indication never results when this unit fails.

The water pump is a poor suspect since the water pump sound is ok.
You have seen that the combustion is abnormal.
Check the units along the path of the oil and the electrical devices.

Figure 3. Answers to *ConsultRu* by *DIAG-orig* and *DIAG-NLP2*

2.1. DIAG-NLP2

In the interest of rapid prototyping, *DIAG-NLP1* was implemented without the benefit of a corpus study. *DIAG-NLP2* is the empirically grounded version of the feedback generator. We collected 23 tutoring interactions between a student using the DIAG tutor on home heating and one of two human tutors. This amounts to 272 tutor turns, of which 235 in reply to *ConsultRU* and 37 in reply to *ConsultInd*. The tutor and the student are in different rooms, sharing images of the same DIAG tutoring screen. When the student consults DIAG, the tutor is provided the same "fact file" that DIAG gives to *DIAG-NLP1*and *DIAG-NLP2*, and types a response that substitutes for DIAG's. The tutor is presented with this information because we wanted to uncover empirical evidence for the aggregation rules to be used in our domain.

We developed a coding scheme [5] and annotated the data. We found that tutors provide explicit problem solving directions in 73% of the replies, and evaluate the student's action in 45% of the replies. As expected, they *exclude* much of the information (63% to be precise) that DIAG would provide, and specifically, always exclude any mention of RUs that are not as likely to cause a certain problem, e.g. the *ignitor assembly* in Fig. 2. Tutors do perform a fair amount of aggregation, as measured in terms of the number of RUs and indicators labelled as *summary*. Further, they use functional, not syntactic or structural, aggregation of parts. E.g., the oil nozzle, supply valve, pump, filter, etc., are described as *the path of the oil flow*.

In *DIAG-NLP2* a planning module manipulates the information given to it by DIAG before passing it to EXEMPLARS, and ultimately to RealPro [9], the sentence realizer that produces grammatical English sentences. This module decides which information to include according to the type of query posed to the system. Here we sketch how the reply at the bottom of Fig. 2 is generated. The planner starts by mentioning the referent of the queried indicator and its state (*The combustion is abnormal*), rather than the indicator itself (this is also based on our corpus study). It then chooses, among all the RUs that DIAG would talk about, only those *REL(evant)-RUs* that would definitely result in the observed symptom. It then decides whether to aggregate them functionally by using a simple heuristics. For each RU, its possible aggregators and the number n of units it covers are listed in a table (e.g., *electrical devices* covers 4 RUs, *ignitor, photoelectric cell, transformer* and *burner motor*). If a group of REL-RUs contains k units covered by aggregator *Agg*, if $k < \frac{n}{2}$, *Agg* will not be used; if $\frac{n}{2} \leq k < n$, *Agg* preceded by *some of* will be used; if $k = n$, *Agg* will be used. Finally, *DIAG-NLP2* instructs the student to

check the possibly aggregated REL-RUs.
Full details on the corpus, the coding scheme and DIAG-NLP2 can be found in a companion paper [3].

3. Experimental Results

Our empirical evaluation is a between-subject study with three groups: the first interacts with *DIAG-orig*, the second with *DIAG-NLP1*, the third with *DIAG-NLP2*. The 75 subjects (25 per group) were all science or engineering majors affiliated with our university. Each subject read some short material about home heating, went through one trial problem, then continued through the curriculum on his/her own. The curriculum consisted of three problems of increasing difficulty. As there was no time limit, every student solved every problem. Reading materials and curriculum were identical in the three conditions.

While a subject was interacting with the system, a log was collected including, for each problem: whether the problem was solved; total time, and time spent reading feedback; how many and which indicators and RUs the subject consults DIAG about; how many, and which RUs the subject replaces. We will refer to all the measures that were automatically collected as *performance measures*.

At the end of the experiment, each subject was administered a post-test, a test of whether subjects remember their actions, and a usability questionnaire.

We found that subjects who used *DIAG-NLP2* had significantly higher scores on the post-test, and were significantly more correct in remembering what they did. As regards performance measures, there are no so clear cut results. As regards usability, subjects prefer the NL enhanced systems to *DIAG-orig*, however results are mixed as regards which of the two they actually prefer.

In the tables that follow, boldface indicates significant differences, as determined by an analysis of variance performed via ANOVA, followed by post-hoc Tukey's tests.

	Post-Test	RU Precision	RU Recall
DIAG-orig	0.72	0.78	0.53
DIAG-NLP1	0.69	0.70	0.47
DIAG-NLP2	**0.90**	**0.91**	0.40

Table 1. Learning Scores

Table 1 reports learning measures, average across the three problems. The post-test consists of three questions and tests what the student has learnt about the domain. Subjects are also asked to remember the RUs they replaced, under the assumption that the better they remember how they solved a certain problem, the better they will be able to apply what they learnt to a new problem - namely, their recollection should correlate with *transfer*. We quantify the subjects' recollec-

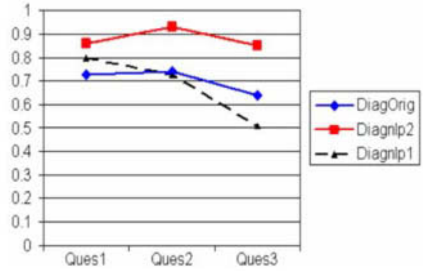


Figure 4.Scores on problems

tions in terms of precision and recall with respect to the log that the system collects. *DIAG-NLP2* is significantly better as regards post-test score ($F = 10.359, p = 0.000$), and RU Precision ($F = 4.719, p = 0.012$).

Performance on individual questions in the post-test is illustrated in Fig. 4. Scores in *DIAG-NLP2* are always higher, significantly so on questions 2 and 3 ($F = 8.481, p = 0.000$, and $F = 7.909, p = 0.001$), and marginally so on question 1 ($F = 2.774, p = 0.069$).

	Time	RU Replaced	ConsultInd	Avg. Time	ConsultRU	Avg. Time
DIAG-Orig	30’17”	8.88	22.16	8”	63.52	5”
DIAG-NLP1	28’34”	11.12	**6.92**	14”	45.68	4”
DIAG-NLP2	34’53"	11.36	28.16	**2"**	52.12	5”

Table 2. Performance Measures across the three systems

Table 2 reports performance measures, cumulative across the three problems (other than average reading times, *Avg. Time*). Subjects don’t differ significantly in the time they spend solving the problems, or in the number of RUs they replace, although they replace fewer parts in *DIAG-orig*. This trend is opposite what we would have hoped for, since when repairing a real system, replacing parts that are working should clearly be kept to a minimum. The simulation though allows subjects to replace as many as they want without any penalty before they come to the correct solution.

The next four entries in Table 2 report the number of queries that subjects ask, and the average time it takes subjects to read feedback from the system. The subjects ask significantly fewer *ConsultInd* in *DIAG-NLP1* ($F = 8.905, p = 0.000$), and take significantly less time reading *ConsultInd* feedback in *DIAG-NLP2* ($F = 15.266, p = 0.000$). The latter result is not surprising, since the feedback in *DIAG-NLP2* is in general much shorter than in *DIAG-orig* and *DIAG-NLP1*. Neither the reason not the significance of subjects asking fewer *ConsultInd* of *DIAG-NLP1* are apparent to us.

We also collected usability measures. Although these are not usually reported in ITS evaluations, in a real setting students should be more willing to sit down with a system that they perceive as more friendly and usable. Subjects rate the system along four dimensions on a five point scale: clarity, usefulness, repetitiveness, and whether it ever misled them (the highest clarity but the lowest repetitiveness receive 5 points). There are no significant differences on individual dimensions. Cumulatively, *DIAG-NLP2* (at 15.08) slightly outperforms the other two (*DIAG-orig* at 14.68 and *DIAG-NLP1* at 14.32), however, the difference is not significant (highest possible rating is 20 points). Finally, on paper, subjects compare two pairs of versions of feedback: in each pair, the first feedback is generated by the system they just worked with, the second is generated by one of the other two systems. Subjects say which version they prefer, and why (they can judge the system along one or more of four dimensions: natural, concise, clear, contentful). In general, subjects prefer the NLP systems to *DIAG-orig* (marginally significant, $\chi^2 = 9.49, p < 0.1$). Subjects find *DIAG-NLP2* more natural, but *DIAG-NLP1* more contentful ($\chi^2 = 10.66, p < 0.025$).[1]

[1] In these last two cases, χ^2 is run on tables containing the number of preferences assigned to each system, in the various categories.

4. Discussion and future work

Only very recently have the first few results become available, to show that first of all, students do learn when interacting in NL with an ITS [6,10,12,14]. However, there are very few studies like ours, that compare versions of the same ITS that differ in specific features of the NL interface. One such study is [10], which found no difference in the learning gains of students who interact with an ITS that tutors in mechanics using typed text or speech.

We did find that different features of the NL feedback impact learning. We claim that the effect is due to using functional aggregation, that stresses an abstract and more conceptual view of the relation between symptoms and faulty parts. However, the feedback in *DIAG-NLP2* changed along two other dimensions: using referents of indicators instead of indicators, and being more strongly directive in suggesting what to do next. Although we introduced the latter in order to model our tutors, it has been shown that students learn best when prompted to draw conclusions by themselves, not when told what those conclusions should be [2]. Thus we would not expect this feature to be responsible for learning.

Naturally, DIAG-NLP2 is still not equivalent to a human tutor. Unfortunately, when we collected our naturalistic data, we did not have students take the post-test. However, performance measures were automatically collected, and they are reported in Table 3 (as in Table 2, measures other than reading times are cumulative across the three problems). If we compare Tables 2 and 3, it is apparent that when interacting with a human tutor, students ask far fewer questions, and they read them much more carefully. The replies from the tutor must certainly be better, also because they can freely refer to previous replies; instead, the dialogue context is just barely taken into account in *DIAG-NLP2* and not taken into account at all in *DIAG-orig* and *DIAG-NLP1*. Alternatively, or in addition, this may be due to the *face* factor [1,11], i.e., one's public self-image: e.g., we observed that some subjects when interacting with any of the systems simply ask for hints on every RU without any real attempt to solve the problem, whereas when interacting with a human tutor they want to show they are trying (relatively) hard. Finally, it has been observed that students don't read the output of instructional systems [8].

Time	RU Replaced	ConsultInd	Avg. Time	ConsultRu	Avg. Time
38'54"	8.1	1.41	21.0"	10.14	14.0"

Table 3. Performance Measures when interacting with human tutors

The DIAG project has come to a close. We are satisfied that we demonstrated that even not overly sophisticated NL feedback can make a difference; however, the fact that *DIAG-NLP2* has the best language and engenders the most learning prompts us to explore more complex language interactions. We are pursuing new exciting directions in a new domain, that of introductory Computer Science, i.e., of basic data structures and algorithms.

Acknowledgments. This work is supported by grants N00014-99-1-0930 and N00014-00-1-0640 from the Office of Naval Research. We are grateful to CoGenTex Inc. for making EXEMPLARS and RealPro available to us.

References

[1] Penelope Brown and Stephen Levinson. *Politeness: Some Universals in Language Usage*. Studies in Interactional Sociolinguistics. Cambridge University Press, 1987.

[2] Michelene T. H. Chi, Stephanie A. Siler, Takashi Yamauchi, and Robert G. Hausmann. Learning from human tutoring. *Cognitive Science*, 25:471–533, 2001.

[3] B. Di Eugenio, D. Fossati, D. Yu, S. Haller, and M. Glass. Aggregation improves learning: experiments in natural language generation for intelligent tutoring systems. In *ACL05, Proceedings of the 42nd Meeting of the Association for Computational Linguistics*, 2005.

[4] M. W. Evens, J. Spitkovsky, P. Boyle, J. A. Michael, and A. A. Rovick. Synthesizing tutorial dialogues. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*, pages 137–140, Hillsdale, New Jersey, 1993. Lawrence Erlbaum Associates.

[5] M. Glass, H. Raval, B. Di Eugenio, and M. Traat. The DIAG-NLP dialogues: coding manual. Technical Report UIC-CS 02-03, University of Illinois - Chicago, 2002.

[6] A.C. Graesser, N. Person, Z. Lu, M.G. Jeon, and B. McDaniel. Learning while holding a conversation with a computer. In L. PytlikZillig, M. Bodvarsson, and R. Brunin, editors, *Technology-based education: Bringing researchers and practitioners together*. Information Age Publishing, 2005.

[7] Susan Haller and Barbara Di Eugenio. Minimal text structuring to improve the generation of feedback in intelligent tutoring systems. In *FLAIRS 2003, the 16th International Florida AI Research Symposium*, St. Augustine, FL, May 2003.

[8] Trude Heift. Error-specific and individualized feedback in a web-based language tutoring system: Do they read it? *ReCALL Journal*, 13(2):129–142, 2001.

[9] Benoît Lavoie and Owen Rambow. A fast and portable realizer for text generation systems. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, 1997.

[10] D. J. Litman, C. P. Rosé, K. Forbes-Riley, K. VanLehn, D. Bhembe, and S. Silliman. Spoken versus typed human and computer dialogue tutoring. In *Proceedings of the Seventh International Conference on Intelligent Tutoring Systems*, 2004.

[11] Johanna D. Moore, Kaska Porayska-Pomsta, Sebastian Varges, and Claus Zinn. Generating tutorial feedback with affect. In *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference*, 2004.

[12] S. Peters, E. Owen Bratt, B. Clark, H. Pon-Barry, and K. Schultz. Intelligent systems for training damage control assistants. In *Proceedings of I/ITSEC 2004, Interservice/Industry Training, Simulation, and Education Conference*, 2004.

[13] Mike Reape and Chris Mellish. Just what *is* aggregation anyway? In *Proceedings of the European Workshop on Natural Language Generation*, Toulouse, France, 1998.

[14] C. P. Rosé, D. Bhembe, S. Siler, R. Srivastava, and K. VanLehn. Exploring the effectiveness of knowledge construction dialogues. In *AIED03, Proceedings of AI in Education*, 2003.

[15] James Shaw. A corpus-based analysis for the ordering of clause aggregation operators. In *COLING02, Proceedings of the 19th International Conference on Computational Linguistics*, 2002.

[16] Douglas M. Towne. Approximate reasoning techniques for intelligent diagnostic instruction. *International Journal of Artificial Intelligence in Education*, 1997.

[17] Michael White and Ted Caldwell. Exemplars: A practical, extensible framework for dynamic text generation. In *Proceedings of the Ninth International Workshop on Natural Language Generation*, pages 266–275, 1998.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Dialogue-Learning Correlations in Spoken Dialogue Tutoring

Kate Forbes-Riley [a] Diane Litman [a] Alison Huettner [a] and Arthur Ward [a]

[a] *University of Pittsburgh, Learning Research and Development Center, 3939 O'Hara Street, Pittsburgh, PA, 15260, USA.*

Abstract. We examine correlations between dialogue characteristics and learning in two corpora of spoken tutoring dialogues: a human-human corpus and a human-computer corpus, both of which have been manually annotated with *dialogue acts* relative to the tutoring domain. The results from our human-computer corpus show that the presence of student utterances that display reasoning, as well as the presence of reasoning questions asked by the computer tutor, both positively correlate with learning. The results from our human-human corpus show that the introduction of a new concept into the dialogue by students positively correlates with learning, but student attempts at deeper reasoning do not, and the human tutor's attempts to direct the dialogue negatively correlate with learning.

1. Introduction

Research in tutorial dialogue systems is founded on the belief that a one-on-one natural language conversation with a tutor provides students with an environment that exhibits characteristics associated with learning. However, it is not yet well understood exactly how specific student and tutor dialogue behaviors correlate with learning, and whether such correlations generalize across different types of tutoring situations.

In the computational tutoring community, understanding such correlations has become of increasing interest, in order to put system building on a more empirical basis; this is because when it comes time to actually implement a tutorial dialogue system, many design choices must be made that will likely influence the style of the dialogue, which in turn may influence a student's ability to learn from the system. One area of interest has been the use of shallow measures to investigate the hypothesis that increased student language production correlates with learning; shallow measures have the advantage of being automatically computable, and are thus easy to incorporate into an online adaptive system. Studies of typed (primarily human-human) dialogue tutoring corpora, for example, have shown that longer student turns, and higher percentages of student words and student turns, all positively correlate with learning [1,2,3].

Unfortunately, when in prior work we applied similar measures to other types of tutoring dialogues - namely *spoken* dialogues, and human-*computer* dialogues (typed and spoken) - we found that although our students learned, most correlations between learning and shallow dialogue measures did not generalize to our data [4]. Furthermore, even when some shallow correlations did generalize (as in our typed human-human data), we felt that further analysis was still needed to better understand our results. For example, one might hypothesize that longer student turns are a good estimate of how much a student explains, but a deeper coding of the data would be needed to test this hypothesis.

In fact, the notion of a "dialogue act" [5,6,7], which attempts to codify the underlying intent behind a student or tutor utterance, has been used in recent studies of both implemented [8] and simulated [9] computer tutors. For example, the correlation studies of [8] suggest that student learning is positively correlated with the use of tutor dialogue acts requiring students to provide the majority of an answer, and negatively correlated with the use of tutor acts where the tutor primarily provides the answer.[1]

In this paper, we take a similar approach, and analyze correlations between learning and dialogue acts. However, we examine learning correlations with both *tutor* as well as *student* dialogue acts. In addition, we examine and contrast our findings across two types of spoken dialogue corpora: one with a *human* tutor, and the other with a *computer* tutor. The results in our human-computer corpus show that the presence of student utterances that display reasoning, as well as the presence of reasoning questions asked by the computer tutor, both positively correlate with learning. The results from our human-human corpus are more complex, mirroring the greater complexity of human-human interaction: the introduction of a new concept into the dialogue by students positively correlates with learning, but student attempts at deeper reasoning do not, and the human tutor's attempts to direct the dialogue can negatively correlate with student learning.

2. Dialogue Data and Coding Schemes

ITSPOKE (**I**ntelligent **T**utoring **SPOKE**n dialogue system) [11] is a *speech-enabled* version of the *text-based* Why2-Atlas conceptual physics tutoring system [12]. Our data consists of two corpora of spoken tutoring dialogues, one with the ITSPOKE computer tutor, and the other with a human tutor performing the same task as ITSPOKE. Both corpora were collected during a prior study [4], using the same experimental procedure: university students 1) took a pretest measuring their physics knowledge, 2) read a small document of background material, 3) used a web and voice interface to work through a set of up to 10 training problems (dialogues) with the tutor, and 4) took a posttest similar to the pretest.[2] In each training problem, students first typed an essay answering a qualitative physics problem; the tutor then engaged the student in spoken dialogue to correct misconceptions and elicit more complete explanations. Annotated (see below) examples from our two corpora are shown in Figures 1 and 2 (punctuation added for clarity).[3]

For our current study, each *tutor turn* and each *student turn* in these two corpora was manually annotated for tutoring-specific dialogue acts.[4] Our tagset of "Student and Tutor Dialogue Acts" is shown and briefly defined in Figure 3. This tagset was developed based on pilot annotation studies using similar tagsets previously applied in other tutorial dialogue projects [13,5,6,7]. As shown, "Tutor and Student Question Acts" label the

[1] Correlations between similar codings of dialogue data have also been studied in collaborative learning research. For example, [10] shows that students who more often indicated that they needed help by asking specific questions learned more than those who asked fewer specific questions ($R= 0.48$, $p < .01$).

[2] In the human-computer corpus, students worked through 5 problems, and took the pretest after the reading.

[3] The *human-computer* corpus contains 100 dialogues (20 students), averaging 22 student turns and 29 tutor turns per dialogue. The *human-human* corpus contains 128 dialogues (14 students), averaging 47 student turns and 43 tutor turns per dialogue.

[4] While one annotator labeled the entire corpus, an agreement study on a subset of the corpus gave 0.67 Kappa and 0.63 Kappa between two annotators on 334 tutor turns and 442 student turns, respectively.

type of question that is asked, in terms of content and the expectation that the content presupposes with respect to the type of answer required. This Act is most common to the tutor; as detailed below, there are no student questions in our human-computer corpus, and they are infrequent in our human-human corpus. "Tutor Feedback Acts" essentially label the "correctness" of the student's prior turn, in terms of explicit positive or negative tutor responses. "Tutor State Acts" serve to summarize or clarify the current state of the student's argument, based on the prior student turn(s). "Student Answer Acts" label the type of answer that a student gives, in terms of the quantity and quality of the content and the extent of reasoning that the content requires. Finally, the "NonSubstantive Act" (NS) tag was used to label turns that did not contribute to the physics discussion (e.g., "Are you ready to begin?").

. . . spoken dialogue excerpt at 2.6 minutes into session. . .

STUDENT$_5$: Gravitational pull, downward. *[NOVS]*

ITSPOKE$_6$: Almost. The forces acting on the pumpkin are both vertical forces. The force exerted by the person is in the upward direction. The person also exerts a force on the pumpkin. One of the forces on the pumpkin is gravity. We have established that there are two forces acting on the pumpkin, gravity in the downward direction and the force exerted by the person in the upward direction. So what is the direction of the NET force? *[BOT, RS, RC, SAQ]*

STUDENT$_7$: Upward. *[NOVS]*

ITSPOKE$_8$: Yeah. Does the pumpkin accelerate during the toss? If yes, please specify the direction of the acceleration. *[POS, SAQ, RD]*

STUDENT$_9$: Yes, upward. *[SHAL]*

ITSPOKE$_{10}$: Well... To analyze the pumpkin's acceleration we will use Newton's second law. What is the definition of this law? *[HNT, LAQ]*

Figure 1. Annotated Human-Computer Dialogue Excerpt

. . . spoken dialogue excerpt at 16.6 minutes into session. . .

TUTOR$_{101}$: Yeah, that's precisely the point. Yes, we are all learning. Ok so uh now uh you apply the same push for the same amount of time for on both the containers then what would you compare to distinguish between them? *[POS, RC, SAQ]*

STUDENT$_{102}$: I would be comparing their rate of velocity. *[NOVS]*

TUTOR$_{103}$: Not rate. You will be comparing their velocity. You see, rate will imply that something is changing which there is no change. Velocity is constant so you will surely compare their velocities. Which one will be faster? *[HNT, RS, SAQ]*

STUDENT$_{104}$: The feathers. *[NOVS]*

TUTOR$_{105}$: The feathers- why? *[RS, DAQ]*

STUDENT$_{106}$: Because there's less matter. *[DEEP]*

Figure 2. Annotated Human-Human Dialogue Excerpt

As Figures 1-2 illustrate, most tutor turns are labeled with multiple Tutor Acts, while most student turns are labeled with a single Student Act. Applying the Dialogue Act coding scheme to our human-computer corpus yielded 2293 Student Acts on 2291 student turns and 6879 Tutor Acts on 2964 tutor turns. Applying the coding scheme to our human-human corpus yielded 5969 Student Acts on 5879 student turns and 7861 Tutor Acts on 4868 tutor turns.

- **Tutor and Student Question Acts**
  - Short Answer Question (**SAQ**): concerns basic quantitative relationships.
  - Long Answer Question (**LAQ**): requires definition/interpretation of concepts.
  - Deep Answer Question (**DAQ**): requires reasoning about causes and/or effects.
- **Tutor Feedback Acts**
  - Positive Feedback (**POS**): overt positive response to prior student turn.
  - Negative Feedback (**NEG**): overt negative response to prior student turn.
- **Tutor State Acts**
  - Restatement (**RS**): repetitions and rewordings of prior student statement.
  - Recap (**RC**): restating student's overall argument or earlier-established points.
  - Request/Directive (**RD**): directions summarizing expectations about student's argument.
  - Bottom Out (**BOT**): complete answer supplied after student answer is incorrect or incomplete.
  - Hint (**HNT**): partial answer supplied after student answer is incorrect or incomplete.
  - Expansion (**EX**): novel details about student answer supplied without first being queried.
- **Student Answer Acts**
  - Deep Answer (**DEEP**): consists of at least two concepts linked together through reasoning.
  - Novel/Single Answer (**NOVS**): consists of one concept introduced by student into dialogue.
  - Shallow Answer (**SHAL**): consists of one concept previously introduced into dialogue.
  - Assertion (**AS**): used for answers such as "I don't know" or equivalent.
- **Tutor and Student Non-Substantive Acts (NS)**: do not contribute to the physics discussion.

Figure 3. Student and Tutor Dialogue Acts

3. Correlation Analysis Methodology

As discussed in Section 1, although our prior work demonstrated that students learned a significant amount with both our human and computer tutors [4], in our spoken data we were unable to find any correlations between learning and a set of shallow dialogue measures of increased student activity (e.g., longer student turns). Here we revisit the question of what aspects of our spoken dialogues correlate with learning, but replace our previous shallow measures for characterizing dialogue with a set of "deeper" measures derived from the Student and Tutor Dialogue Act annotations described in Section 2.

For each of our two corpora, we first computed for each student, a total, a percentage, and a ratio representing the usage of each Student and Tutor Dialogue Act tag across all of the dialogues with that student. We call these measures our *Dialogue Act Measures*. Each *Tag Total* was computed by counting the number of (student or tutor) turns that contained that tag at least once. Each *Tag Percentage* was computed by dividing the tag's total by the total number of (student or tutor) turns. Finally, each *Tag Ratio* was computed by dividing the tag's total by the total number of (student or tutor) turns that contained a tag of that tag *type*. For example, suppose the dialogue in Figure 1 constituted our entire corpus. Then our Dialogue Act Measures for the Tutor "POS" tag would be: Tag Total = 1, since 1 tutor turn contains the "POS" tag. Tag Percentage = 1/3, since there are 3 tutor turns. Tag Ratio = 1/1, since 1 tutor turn contains a Tutor Feedback Act tag.

Next, for each of the Dialogue Act Measures, we computed a Pearson's correlation between the measure and posttest score. However, because the pretest and posttest scores were significantly correlated in both the human-human (R=.72, p =.008) and human-computer corpora (R=.46, p=.04), we controlled for pretest score by regressing it out of the correlation.[5] In the following Sections (4 and 5), we present and discuss the best results of these correlation analyses, namely those where the correlation with learning was significant ($p \leq .05$) or a trend ($p \leq .1$), after regressing out pretest.

4. Human-Computer Results

Table 1 presents our best results on correlations between Dialogue Act Measures and learning in our human-computer corpus. The first column lists the measure (total (#), percentage (%) or ratio (Rat:) of the Dialogue Act per student). The second and third columns show the mean and standard deviation (across all students), while the last two columns present the Pearson's correlation between posttest and the measure after the correlation with pretest is regressed out. For example, the first row shows that there are 11.90 total Deep Answers over all the dialogues of a student on average, and that there is a statistically significant (p=.04) positive correlation (R = .48) between total Deep Answers and posttest, after the correlation with pretest is regressed out.

Table 1. Dialogue-Learning Correlations: Human-Computer Corpus (20 students)

Dialogue Act Measure	Mean	Std.Dev.	R	p
# Student DEEP	11.90	5.78	.48	.04
# Tutor DAQ	9.59	4.89	.41	.08
% Tutor DAQ	6.27%	2.30%	.45	.05
% Tutor Question Acts	76.89%	3.12%	.57	.01
Rat: Tutor SAQ to Question Acts	.88	.04	-.47	.04
Rat: Tutor DAQ to Question Acts	.08	.03	.42	.07
# Tutor POS	76.10	16.66	.38	.10

As shown, the *type of answer provided by students* relates to how much they learn in our human-computer corpus, as indicated by the positive correlation between student Deep Answers and learning. Note that there are no significant (positive or negative) correlations for student Shallow or Novel/Single Answers, or a student's inability to provide an answer (Assertions), which suggests that the relationship between student answer type and learning requires further analysis.

The *type of questions asked by tutors* also relates to how much students learn in our human-computer corpus. There is a positive correlation between the percent of tutor Deep Answer Questions and learning, and a trend for the number and ratio of tutor Deep Answer Questions to positively correlate with learning. In contrast, there is a negative correlation between the ratio of tutor Short Answer Questions and learning. The *quantity of tutor questions* also relates to student learning, as evidenced by the strong positive correlation between the overall percentage of all tutor Question Acts and learning.

Table 1 also shows a slight trend for tutor Positive Feedback to positively correlate with learning. Other studies have shown positive relationships between encouragement during computer tutoring and student outcomes [14]. Finally, note that none of the tutor

[5] The human-human means for the (multiple-choice) pre- and posttests were 0.42 and 0.72, respectively, and the human-computer means were 0.48 and 0.69, respectively.

State Acts correlated with learning, suggesting that the best way to use such organizational acts is not yet fully understood in our computer tutor.

5. Human-Human Results

Table 2 presents our best results on correlations between Dialogue Act Measures and learning in our human-human corpus, using the same format as Table 1. As shown, the *type of dialogue acts used by students* relates to how much students learn in our human-human corpus too. With respect to student answers, here we find a trend for the number and ratio of student Novel/Single Answers to positively correlate with learning; however, in contrast to our human-computer results, we also find a trend for the number of student Deep Answers to *negatively* correlate with learning. Moreover, unlike in the human-computer corpus, in our human-human corpus students do ask questions. Here we see that a higher ratio of student Short Answer Questions positively correlates with learning, and a higher ratio of student Long Answer Questions negatively correlates with learning.

Table 2. Dialogue-Learning Correlations: Human-Human Corpus (14 students)

Dialogue Act Measure	Mean	Std.Dev.	R	p
# Student NOVS	19.29	7.95	.49	.09
# Student DEEP	68.50	27.99	-.49	.09
Rat: Student NOVS to Answers	.14	.05	.47	.10
Rat: Student SAQ to Question Acts	.91	.12	.56	.05
Rat: Student LAQ to Questions	.03	.08	-.57	.04
# Tutor RD	19.86	10.58	-.71	.01
% Tutor RD	5.65%	.02	-.61	.03
# Tutor RS	79.14	26.83	-.56	.05
# Tutor NEG	14.50	7.60	-.60	.03

Table 2 also shows that the *type of dialogue acts used by the tutor* relates to how much students learn in our human-human corpus. In contrast to the human-computer corpus, in our human tutoring dialogues we only find correlations with non-question tutor Acts (namely State Acts and Negative Feedback), and also find only negative correlations. The correlations between tutor State Acts (RD, RS) and learning show that increased tutor summarization and clarification negatively correlates with student learning. We also see a negative correlation between tutor Negative Feedback and learning.

6. Discussion

Our human-human corpus represents an upper bound for the speech and natural language processing capabilities of our ITSPOKE corpus. As such, cross-corpora differences in how student and tutor dialogue acts relate to student learning can shed light on how system improvements might positively impact learning. We see little overlap in terms of the correlations between tutoring Dialogue Acts and learning across our human-computer and human-computer corpora. In our computer tutoring data, we found that student learning was positively correlated with both the presence of student utterances displaying reasoning, as well as the presence of tutor questions requiring reasoning. These results are similar to previous findings in human-tutoring data, where learning was correlated with both students' construction of knowledge, and tutor behaviors prompting students to construct knowledge [13]. We hypothesize that because Deep Answers involve more student reasoning, they involve more knowledge construction. Note that we previously found no significant correlation between average turn length (# words/turn) or dialogue length (total words) and learning in either our human-computer or human-human corpora [4]; together these results suggest that it is not the quantity but the quality of the students' responses that correlate with learning.

The results from our human-human corpus are more complex. First, there is no longer a straightforward correlation between the depth of reasoning displayed in student answers and learning: while student Novel/Single insights positively correlate with learning, student attempts at even deeper reasoning negatively correlate with learning. While this negative correlation is surprising, inspection of the student turns in the human-human corpus leads us to hypothesize that student Deep Answers might often be incorrect, which itself might negatively correlate with learning, and may also be related to the fact that in the human-human corpus, students speak longer and more freely than in the human-computer corpus. We are currently annotating "correctness", to investigate whether more Deep Answers are "incorrect" or "partially correct" in the human tutoring corpus compared to the computer tutoring corpus, and whether the number of correct answers positively correlates with learning. Similarly, the correlations between tutor Feedback and learning in both corpora might also reflect correctness. Second, while student question-asking is often considered a constructive activity [13], we similarly did not see a straightforward relation between question-asking and learning: while student Short Answer Questions positively correlate with learning, student Long Answer Questions negatively correlate. However, there were only 12 Long Answer Questions in our human-human data, and all displayed clear evidence of student misunderstanding (e.g., containing phrases such as "what do you mean?"). Finally, although we find negative correlations between learning and tutor State Acts (e.g., involving summarization and clarification), attributing any causal relationship would require further research.

Finally, we see some overlap between our results and those of [8], who computed correlations between student learning and tutor dialogue acts in the AutoTutor system. [8] found that students who received more "Hints" (which require the student to provide most of the answer) learned more than those who received more "Assertions" (in which the tutor provides most of the answer). Although our Tutor Act coding is not identical, our "Bottom Out" largely corresponds to their "Assertion"; in our human-human corpus there was a non-significant negative correlation (R=-.00,p=.99), but in our human-computer corpus there was a non-significant positive correlation (R=.08, p=.75), with learning. Our "Hint" is similar to their "Hint"; in our human-computer corpus there was a non-significant positive correlation (R=.26, p=.28), but in our human-human corpus there was a non-significant negative correlation (R=-.38,p=.20), with learning.

7. Conclusions and Current Directions

This paper presented our findings regarding the correlation of student and tutor dialogue acts with learning, in both human-human and human-computer spoken tutoring dialogues. Although we found significant correlations and trends in both corpora, the results for specific dialogue acts differed. This suggests the importance of training systems from appropriate data. The results in our human-computer corpus show that student utterances that display reasoning, as well as tutor questions that ask for student reasoning, both positively correlate with learning. The results in our human-human corpus mirror the greater complexity of human-human interaction: student novel insights positively correlate with learning, but student deeper reasoning is negatively correlated with learning, as are some of the human tutor's attempts to direct the dialogue. As noted above, to gain further insight into our results, we are currently annotating our dialogues for correctness. This will allow us to test our hypothesis that student deep reasoning is more error-prone in the human-human corpus. We are also investigating correlations between learning and *patterns* of dialogue acts, as found in multi-level coding schemes such as [7].

Acknowledgments

We thank Mihai Rotaru and Pam Jordan for their help in improving this paper. This research is supported by ONR (N00014-04-1-0108) and NSF (0325054).

References

[1] M. G. Core, J. D. Moore, and C. Zinn. The role of initiative in tutorial dialogue. In *Proc. European Chap. Assoc. Computational Linguistics*, 2003.

[2] C. P. Rosé, D. Bhembe, S. Siler, R. Srivastava, and K. VanLehn. The role of why questions in effective human tutoring. In *Proceedings of Artificial Intelligence in Education*, 2003.

[3] Sandra Katz, David Allbritton, and Johen Connelly. Going beyond the problem given: How human tutors use post-solution discussions to support transfer. *International Journal of Artificial Intelligence and Education*, 13, 2003.

[4] D. J. Litman, C. P. Rose, K. Forbes-Riley, K. VanLehn, D. Bhembe, and S. Silliman. Spoken versus typed human and computer dialogue tutoring. In *Proc. Intell. Tutoring Systems*, 2004.

[5] A. Graesser and N. Person. Question asking during tutoring. *American Educational Research Journal*, 31(1):104–137, 1994.

[6] A. Graesser, N. Person, and J. Magliano. Collaborative dialog patterns in naturalistic one-on-one tutoring. *Applied Cognitive Psychology*, 9:495–522, 1995.

[7] R. M. Pilkington. Analysing educational discourse: The DISCOUNT scheme. Computer-Based Learning Unit 99/2, University of Leeds, 1999.

[8] G. Jackson, N. Person, and A. Graesser. Adaptive tutorial dialogue in AutoTutor. In *Proc. Workshop on Dialog-based Intelligent Tutoring Systems at Intelligent Tutoring Systems*, 2004.

[9] M. Wolska, B. Q. Vo, D. Tsovaltzi, I. Kruiff-Korbayová, E. Karagjosova, H Horacek, A. Fiedler, and C. Benzmuller. An annotated corpus of tutorial dialogs on mathematical theorem proving. In *Proc. Language Resources and Evaluation*, 2004.

[10] N. Webb and A. M. Mastergeorge. The development of student helping behavior and learning in small groups. *Cognition and Instruction*, 21(4):361–428, 2003.

[11] D. Litman and S. Silliman. ITSPOKE: An intelligent tutoring spoken dialogue system. In *Companion Proc Human Language Technology: North American Chap. Assoc. Computational Linguistics*, 2004.

[12] K. VanLehn, P. W. Jordan, C. P. Rosé, D. Bhembe, M. Böttner, A. Gaydos, M. Makatchev, U. Pappuswamy, M. Ringenberg, A. Roque, S. Siler, R. Srivastava, and R. Wilson. The architecture of Why2-Atlas: A coach for qualitative physics essay writing. In *Proc. Intelligent Tutoring Systems*, 2002.

[13] M. T. H. Chi, S. A. Siler, H. Jeong, T. Yamauchi, and R. G. Hausmann. Learning from human tutoring. *Cognitive Science*, 25:471–533, 2001.

[14] G. Aist, B. Kort, R. Reilly, J. Mostow, and R. Picard. Experimentally augmenting an intelligent tutoring system with human-supplied capabilities: Adding Human-Provided Emotional Scaffolding to an Automated Reading Tutor that Listens. In *Proc. Intelligent Tutoring Systems*, 2002.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Adolescents' Use of SRL Behaviors and Their Relation to Qualitative Mental Model Shifts While Using Hypermedia

Jeffrey A. GREENE and Roger AZEVEDO
University of Maryland, Department of Human Development, 3304 Benjamin Building, 3304E, College Park, MD, 20742, USA, E-mail: jgreene@umd.edu; razevedo@umd.edu

Abstract: This study examined 214 adolescents' use of self-regulated learning (SRL) behaviors when learning about the circulatory system using hypermedia for 40 minutes. We examined students' verbal protocols to determine the relationship between SRL behaviors and qualitative shifts in students' mental models from pretest to posttest. Results indicated that students who exhibited a qualitative shift in their mental models pre to posttest displayed differential use of six SRL behaviors. These SRL behaviors included metacognitive monitoring activities and learning strategies. Implications for the design of hypermedia learning environments are presented.

1. Introduction

When using hypermedia learning environments to study complex and challenging science topics such as the circulatory system, students must regulate their learning [1, 2, 3]. Complex science topics have many characteristics that make them difficult to understand [4, 5, 6, 7]. For example, in order to have a coherent understanding of the circulatory system, a learner must comprehend an intricate system of relations that exist at the molecular, cellular, organ, and system-levels [5, 8, 9]. Understanding system complexity is sometimes difficult because the properties of the system are not available for direct inspection. In addition, students must integrate multiple representations (e.g., text, diagrams, animations) to attain a fundamental conceptual understanding and then use those representations to make inferences [10]. These inferences and mental representations combine to form a learner's mental model of the system. In this study, we focus on students' use of specific regulatory processes during learning with hypermedia and how those processes are related to qualitative shifts in students' mental models of the circulatory system.

Recently, some researchers and educators have turned to hypermedia learning environments as a potential means of enhancing students' understanding of complex science topics such as the circulatory system, ecology, and evolution [6, 7]. There is, however, a continuing debate about the effectiveness of such technologies for learning. Several cognitive and educational researchers (e.g., [11, 12]) have recently begun to empirically test the effectiveness of hypermedia environments on students' learning. This research addressed several cognitive issues related to learning, including the roles of basic cognitive structures in comprehension, the role of underlying cognitive processes in learning from dynamic representations, the role of multiple representations (e.g., text, diagrams, animations), navigation profiles based on learners' trace history, and the manipulation of hypermedia system structure (e.g., linear vs. hierarchical) and embedded features (e.g., advance organizers) designed to foster learning. While these studies have begun to examine several aspects of learning with hypermedia, very few have attempted to examine the role of self-regulation during learning with hypermedia (e.g., [13]).

Self-regulated learning (SRL) is an active, constructive process whereby learners set goals for their learning and then attempt to monitor, regulate, and control their cognition, motivation, and behavior in the services of those goals [14]. Thus, self-regulated learners efficiently manage their own learning in many different ways [15]. SRL is guided and constrained by both personal characteristics and the contextual features of the environment [14]. Based on models of self-regulation (e.g., [14, 16, 17]), self-regulating students using hypermedia environments to learn about complex and challenging science topics typically engage in a series of recursive cycles of cognitive activities central to learning and knowledge construction activities (e.g., [1, 2]). Specifically, they may initially plan and set goals based on their previous experience with similar tasks. They may engage in various monitoring processes that represent metacognitive awareness of different aspects of the self, task, and context. They may use strategies to control and regulate different aspects of the self, task, learning environment, and learning context. Furthermore, they may also experience various kinds of reactions to and reflections of themselves as learners (e.g., feel more efficacious because of their success in using a particular learning strategy), the task, and context.

Little research has been conducted to examine the interrelatedness and dynamics of SRL variables—cognitive, motivational/affective, behavioral, and contextual—during the cyclical and iterative phases of planning, monitoring, control, and reflection that take place when students learn with hypermedia environments (e.g., [13]). The question of how students regulate their learning about complex science topics while using hypermedia environments is critical to examining how these SRL variables facilitate students' learning. One means of assessing learning in this context is to measure shifts in students' mental models from pretest to posttest. We argue that the presence or absence of qualitative shifts in students' mental models of the circulatory system is related to the specific type and the frequency of use of SRL variables.

In other words, different qualitative shifts in conceptual understanding, as measured by pretest-posttest mental shifts during learning with hypermedia, are based on the use of specific SRL variables and their frequency during learning. For example, previous research indicates that the use of feeling of knowing (FOK) as a metacognitive monitoring activity is related to enhanced understanding of the circulatory system (e.g., [1]). By contrast, the use of control of context as a metacognitive monitoring activity used by learners is associated with lower shifts in students' mental models because it involves monitoring several aspects of the hypermedia learning environment instead of focusing on learning [1]. In this study, we examined which SRL variables are responsible for the presence or absence of qualitative changes in students' mental model shifts of the circulatory system. Examining the frequency of use of these SRL variables in relation to qualitative shift in students' mental models is a critical contribution toward informing the design of hypermedia learning environments [7].

2. Method

2.1 Participants. Participants were 214 middle school and high school students (MS N=113; HS N=101) located outside a large mid-Atlantic city in the United States of America. The mean age of the middle school subjects was 12 years (*SD* = 1) and the mean age of the high school students was 15 years (*SD* = 1)..

2.2 Measure. The paper-and-pencil materials consisted of a consent form, a participant questionnaire, a pretest, and a posttest. All of the paper-and-pencil materials were constructed in consultation with a nurse practitioner who is a faculty member at a school of nursing in a large mid-Atlantic university and a science teacher.

Both the pretest and the posttest consisted of a sheet that contained the instruction, *"Please write down everything you can about the circulatory system. Be sure to include all the parts and their purpose, explain how they work both individually and together, and also explain how they contribute to the healthy functioning of the body".* The posttest was identical to the pretest.

2.3 Hypermedia Learning Environment (HLE). During the experimental phase, the participants used a HLE to learn about the circulatory system. In this HLE, the circulatory system is covered in three articles, comprised of 16,900 words, 18 sections, 107 hyperlinks, and 35 illustrations. The HLE included a table of contents for each article and both global and local search functions.

2.4 Procedure. The authors, along with three trained graduate students, tested participants individually in all conditions. Informed consent was obtained from all participants' parents. First, the participant questionnaire was handed out, and participants were given as much time as they wanted to complete it. Second, the pretest was handed out, and participants were given 20 minutes to complete it. Participants wrote their answers on the pretest and did not have access to any instructional materials. Third, the experimenter provided instructions for the learning task. The following instructions were read and presented to the participants in writing.

"*You are being presented with a hypermedia learning environment, which contains textual information, static diagrams, and a digitized video clip of the circulatory system. We are trying to learn more about how students use hypermedia environments to learn about the circulatory system. Your task is to learn all you can about the circulatory system in 40 minutes. Make sure you learn about the different parts and their purpose, how they work both individually and together, and how they support the human body. We ask you to 'think aloud' continuously while you use the hypermedia environment to learn about the circulatory system. I'll be here in case anything goes wrong with the computer or the equipment. Please remember that it is very important to say everything that you are thinking while you are working on this task."*

Following the instructions, a practice task was administered to encourage all participants to give extensive self-reports on what they were inspecting and reading in the hypermedia environment and what they were thinking about as they learned. During the learning task, an experimenter remained nearby to remind participants to keep verbalizing when they were silent for more then three seconds (e.g., *"Say what you are thinking"*). All participants were reminded of the global learning goal (*"Make sure you learn about the different parts and their purpose, how they work both individually and together, and how they support the human body"*) as part of their instructions for learning about the circulatory system. All participants had access to the instructions (which included the learning goal) during the learning session. All participants were given 40 minutes to use the hypermedia environment to learn about the circulatory system.

All participants were given the posttest after using the hypermedia environment to learn about the circulatory system. They were given 20 minutes to complete the posttest by writing their answers on the sheets provided by one of the experimenters. All participants independently completed the posttest in 20 minutes without their notes or any other instructional materials.

2.5 Coding and Scoring. In this section we describe the coding of the students' mental models. Our analyses focused on the shifts in participants' mental models. One goal of our research was to capture each participant's initial and final mental model of the circulatory system. This analysis depicted the status of each student's mental model prior to and after learning as an indication of representational change that occurred during learning. In our case, the status of the mental model refers to the correctness and completeness in regard to the local features of each component, the relationships between and among the local features of each component, and the relationships among the local features of different components.

We followed Azevedo and colleagues' [1, 2, 13] method for analyzing the participants' mental models, which is based on Chi and colleagues' research [5, 8, 9]. A student's initial mental model of how the circulatory system works was derived from their statements on the pretest essay. Similarly, a student's final mental model of how the circulatory system works was derived from their statements from the essay section of the posttest. Azevedo and colleagues' scheme consists of 12 mental models which represent the progression from no understanding to the most accurate understanding: (a) no understanding, (b) basic global concept, (c) basic global concept with purpose, (d) basic single loop model, (e) single loop with purpose, (f) advanced single loop model, (g) single loop model with lungs, (h) advanced single loop model with lungs, (i) double loop concept, (j) basic double loop model, (k) detailed double loop model, and (l) advanced double loop model. The mental models are based on biomedical knowledge provided by the consulting nurse practitioner**.** For a complete description of the necessary features for each mental model see Azevedo and Cromley [1, p. 534-535].

The second author and a trained graduate student scored the students' pretest and posttest mental models by assigning the numerical value associated with the mental models described in Azevedo and Cromley [1]. For example, a student who began by stating that blood circulates would be given a mental model of "b". If that same student on the posttest also described the heart as a pump, mentioned blood vessel transport, described the purpose of the circulatory system, and included details about blood cells or named specific vessels in the heart, he or she would be given a mental model of "f". The values for each student's pretest and posttest mental model were recorded and used in a subsequent analysis to determine the shift in their conceptual understanding (see inter-rater agreement below).

Mental model pre and post test scores were used to determine subject categorization. Consultation with a graduate student and a science teacher led to the determination of two qualitative, as opposed to quantitative, shifts in the mental model rubric. The shift from (f) to (g) was deemed an important qualitative change in students' understanding because (g) introduces the lungs as a vital part of the circulatory system. Thus, any student scoring (f) or below was placed into the "low understanding" category. The other significant qualitative shift in understanding was determined to occur between (h) and (i), due to (i) introducing the concept of a double loop. Therefore students scoring either (g) or (h) were placed in the "medium understanding" group, while students scoring (i) or above were placed in the "high understanding group". Thus, each student had two designations: one for the student's pretest designation (low, medium, or high) and one for the posttest designation (low, medium, or high). In addition, we classified all students whose posttest score was lower than their pretest score as "negative shift" students. It is not clear why some students scored lower on their posttest than on their pretest, suggesting that this is an issue for future research. In sum, there were seven designations for students' mental model performance pre to post test: low/low, low/medium, low/high, medium/medium, medium/high, high/high, and negative shift (see Table 1). Only three of these designations represented a qualitative mental model shift (low/medium, low/high, and medium/high).

Table 1: Mental Model Shift Group Classifications

Mental Model Shift Classification	Mental Model Pretest Score	Mental Model Posttest Score
Low/low	a-f	a-f
Low/medium	a-f	g-h
Low/high	a-f	g-h
Medium/Medium	g-h	g-h

Medium/High	g-h	h-i
High/High	h-i	h-i
Negative Shift	Any case where posttest > pretest	

2.6 Students' verbalizations. The raw data collected from this study consisted of 8560 minutes (142.7 hours) of audio and video tape recordings from 214 participants, who gave extensive verbalizations while they learned about the circulatory system. During the first phase of data analysis, a graduate student transcribed the audio tapes and created a text file for each participant. This phase of the data analysis yielded a corpus of 3047.9 single-spaced pages (*M* = 14.2 pages per participant) with a total of 926,724 words (*M* = 4331 words per participant) A second graduate student verified the accuracy of the transcriptions by comparing each text file with the video tape recording of the participant and no changes were made to the original files. This process was critical in order for the experimenters to later code the learners' SRL behavior.

2.7 Learners' regulatory behavior. Azevedo and colleagues' [1, 2, 13] model of SRL was used to analyze the learners' regulatory behavior. Their model is based on several recent models of SRL [14, 15, 16, 17]. It includes key elements of these models (i.e., Winne's [16] and Pintrich's [14] formulation of self-regulation as a four-phase process), and extends these key elements to capture the major phases of self-regulation. These are: (a) planning and goal setting, activation of perceptions and knowledge of the task and context, and the self in relationship to the task; (b) monitoring processes that represent metacognitive awareness of different aspects of the self, task, and context; (c) efforts to control and regulate different aspects of the self, task, and context; and, (d) various kinds of reactions and reflections on the self and the task and/or context. Azevedo and colleagues' model also includes SRL variables derived from students' self-regulatory behavior that are specific to learning with a hypermedia environment (e.g., selecting a new informational source).

The classes, descriptions and examples from the think-aloud protocols of the planning, monitoring, strategy use, task difficulty and demands, and interest variables used for coding the learners' and tutor's regulatory behavior are presented in Azevedo and Cromley [1, p. 533-534]. We used Azevedo and colleagues' SRL model to re-segment the data from the previous data analysis phase. This phase of the data analysis yielded 25715.7 segments (*M* = 120.2 per participant) with corresponding SRL variables. A trained graduate student used the coding scheme and coded all of the transcriptions by assigning each coded segment with one of the 35 SRL variables.

2.8 Scoring. SRL behaviors coded from the transcripts were tallied for each individual student. Median instances of each SRL behavior across students were determined, and each student was designated as having exhibited each SRL behavior either above or below the median. For example, across all subjects the median percent of total SRL behaviors devoted to prior knowledge activation (PKA) was 4%. Thus, any student whose number of PKA behaviors were less than 4% of that student's total SRL behaviors was classified as using PKA below the median. Likewise, those students who engaged in PKA enough to account for more than 4% of their total SRL behaviors were classified as being above the median.

2.9 Inter-rater agreement. Inter-rater agreement was established by having the graduate student with external training use the description of the mental models developed by Azevedo and colleagues [1, 2, 13]. They independently coded all selected protocols (pre- and posttest essays of the circulatory system from each participant). There was agreement on 415 out of a total of 428 student descriptions, yielding an inter-rater agreement of .97. Inter-rater agreement was also established for the coding of the learners' behavior by comparing the individual coding of the first author with that of the second author. The first author independently re-coded all 25715.7protocol segments (100%).

There was agreement on 24944.2 out of 25715.7 segments yielding an inter-rater agreement of .97. Inconsistencies were resolved through discussion among the co-authors and graduate students.

3. Results

Several two-way contingency table analyses were produced to determine whether students who exhibited a qualitative shift in their mental model pre to posttest (low/medium, low/high, or medium/high) tended to utilize SRL variables at a frequency above the median more than students who showed no qualitative shift or who had a negative shift. A series of chi-square analyses were performed for each SRL variable, with the subject-grouping variable being their qualitative shift designation (low/low, low/medium, low/high, medium/medium, medium/high, high/high, or negative shift). Of the 35 SRL variables examined, 6 were significantly related to qualitative shift category: Feeling of Knowing (FOK), Inference (INF), Knowledge Elaboration (KE), Prior Knowledge Activation (PKA), Re-reading (RR), and Summarization (SUM).

3.1 Feeling of Knowing. FOK is a metacognitive monitoring activity that involves the learner checking whether he or she has sufficiently learned the material in question. FOK and mental model shift group were found to be significantly related, Pearson χ^2 (6, N = 214) = 20.440, p = .002. Cramér's V = .309. Frequencies by qualitative shift group are shown below (see Table 2). These data suggest that more frequent use of FOK is associated with students making a qualitative shift from one mental model category to another.

3.2 Inference. This SRL is a learning strategy that involves students making an inference based upon what was viewed in the HLE. INF and qualitative shift group were found to be significantly related, Pearson χ^2 (6, N = 214) = 19.378, p = .004. Cramér's V = .301. Frequencies by mental model shift group are shown below (see Table 2). These data suggest that students who experience a shift in mental model are also more likely to be above the median in their use of inference.

3.3 Knowledge Elaboration. KE is a learning strategy that involves the student elaborating on material within the HLE. KE and mental model shift group were found to be significantly related, Pearson χ^2 (6, N = 214) = 14.293, p = .027. Cramér's V = .258. Frequencies by mental model shift group are shown below (see Table 2). These data are interesting in that the no shift groups had a larger number of students above the median. However, the results indicate that the three shift groups that experienced qualitative change in mental model pre to posttest (Low/medium, Low/high, Medium/high) had a much more even split between above and below median students. This suggests that while significantly less than 50% of all students engaged in KE, the shift groups experiencing qualitative change had proportionally more of their students above the median.

3.4 Prior Knowledge Activation. Students use PKA to search their memory for relevant information related to the current learning task. PKA and mental model shift group were found to be significantly related, Pearson χ^2 (6, N = 214) = 14.420, p = .025. Cramér's V = .260. Frequencies by qualitative shift group are shown below (see Table 2). These data suggest that more frequent use of FOK is associated with students who made a qualitative shift from one mental model category to another.

3.5 Re-read. RR is a learning strategy that involves having the student go back to read a section of the HLE already covered. RR and mental model shift group were found to be significantly related, Pearson χ^2 (6, N = 214) = 16.207, p = .013. Cramér's V = .275. Frequencies by qualitative shift group are shown below (see Table 2). These data suggest that engaging in re-reading above the median is more common in shift groups that did not experience positive mental model shift.

3.6 Summarization. Students who use SUM as a learning strategy go back and rephrase read or learned material in their own words. SUM and mental model shift group were found to be significantly related, Pearson χ^2 (6, N = 214) = 15.829, p = .015. Cramér's V = .272. Frequencies by qualitative shift group are shown below (see Table 2). These data suggest that in the groups experiencing positive shift pre to posttest, the use of summarization is more often above the median.

3.7 Summary of Results. There data provide a unique perspective upon how students use HLEs to learn about complex and challenging science topics. Overall, these data make the case that students in middle and high school who are engaged in more monitoring of their understanding, through the use of FOK and PKA, are also associated with a higher proportion of qualitative shifts in understanding. In addition, certain strategies, such as inference, knowledge elaboration, and summarization, seem to be associated with positive mental model shifts. Thus, hypermedia environments that promote these SRL behaviors would seem to be more likely to elicit qualitative mental model shift. Re-reading, on the other hand, would seem to be an indicator of student difficulty with the material, and might be a cue for the environment to review recently presented material, as opposed to moving on to another task.

Table 2: Results for the Comparison of SRL Behavior Median Split by Mental Model Shift Group

	Low/Low	Low/Medium	Low/High	Medium/Medium	ım/High	High/High	Negative Shift
FOK > Median	26	12	25	6		6	16
FOK < Median	45	11	9	7		2	25
INF > Median	30	13	24	8		3	13
INF < Median	41	10	10	6		5	28
KE > Median	15	11	16	3		3	9
KE < Median	56	12	18	11		5	32
PKA > Median	35	12	19	5		6	13
PKA < Median	36	11	15	9		2	28
RR > Median	43	12	11	6		4	24
RR < Median	28	11	23	8		4	17
SUM > Median	33	10	20	8		2	16
SUM < Median	38	13	14	6		6	25

4. Implications for the Design of Hypermedia

This research can inform the design and use of HLEs with complex science topics such as the circulatory system. The tremendous opportunities afforded to educators through the use of HLEs will only come to fruition if these learning environments are built to scaffold higher-order learning behaviors. This study points to the importance of creating HLEs that are clear in their presentation, lest unnecessary re-reading of the material take time away from higher order student cognition and learning. On the other hand, it would seem that higher order cognitive strategies, such as summarization and knowledge elaboration, are more likely to lead to the types of qualitative mental model shifts that are essential for true understanding. HLEs could scaffold these strategies by providing prompts and examples of these behaviors. Likewise, students should be encouraged to monitor their understanding both through the activation of prior knowledge and through checking their learning through FOK. HLEs should also prompt such behaviors, perhaps through asking thought questions at the beginning of the section, and presenting mini-quizzes when students proceed to the end of that section. Truly adaptive HLEs would use student trace logs to adaptively approximate students' dynamically changing mental models, providing the necessary feedback to put struggling students back on track while helping successful students achieve new heights [18, 19]. The practical applications of this research lie in the design of HLEs that both decrease the need for lower-level SRL behaviors such as re-reading and increase the use of higher-order ones such as FOK. More research remains to be done, however, on how these higher-order SRL behaviors can be prompted and taught during student use of HLEs. Future research should focus on the best means of inculcating effective SRL behaviors through on-line methods, so that HLEs can teach both content and the actual process of learning.

5. Acknowledgements

This research was supported by funding from the National Science Foundation (REC#0133346) awarded to the second author. The authors would like the thank Jennifer Cromley, Fielding Winters, Daniel Moos, and Jessica Vick for assistance with data collection.

6. References

[1] Azevedo, R., & Cromley, J.G. (2004). Does training on self-regulated learning facilitate students' learning with hypermedia? *Journal of Educational Psychology, 96*(3), 523-535.

[2] Azevedo, R., Cromley, J.G., & Seibert, D. (2004). Does adaptive scaffolding facilitate students' ability to regulate their learning with hypermedia? *Contemporary Educational Psychology, 29*, 344-370.

[3] Shapiro, A., & Niederhauser, D. (2004). Learning from hypertext: Research issues and findings. In D. H. Jonassen (Ed.). *Handbook of Research for Education Communications and Technology (2nd ed).* Mahwah, NJ: Lawrence Erlbaum Associates.

[4] Azevedo, R., Winters, F.I., & Moos, D.C. (in press). Can students collaboratively use hypermedia to learn about science? The dynamics of self- and other-regulatory processes in an ecology classroom. *Journal of Educational Computing Research.*

[5] Chi, M. T.H., Siler, S., Jeong, H., Yamauchi, T., & Hausmann, R. (2001). Learning from human tutoring. *Cognitive Science, 25*, 471-534.

[6] Jacobson, M., & Kozma, R. (2000). *Innovations in science and mathematics education: Adavnced designs for technologies of learning*. Mawah, NJ: Erlbaum.

[7] Lajoie, S.P., & Azevedo, R. (in press). Teaching and learning in technology-rich environments. In P. Alexander, P. Winne, & G. Phye (Eds.), *Handbook of educational psychology* (2nd ed.). Mahwah, NJ: Erlbaum.

[8] Chi, M. T.H., de Leeuw, N., Chiu, M.-H., & LaVancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science, 18*, 439-477.

[9] Chi, M. T.H., Siler, S., & Jeong, H. (2004). Can tutors monitor students' understanding accurately? *Cognition and Instruction, 22*, 363-387.

[10] Kozma, R., Chin, E., Russell, J., & Marx, N. (2000). The roles of representations and tools in the chemistry laboratory and their implications for chemistry learning. *Journal of the Learning Sciences,* 9(2), 105-144.

[11] Jacobson, M., & Archodidou, A. (2000). The design of hypermedia tools for learning: Fostering conceptual change and transfer of complex scientific knowledge. *Journal of the Learning Sciences, 9*(2), 149-199.

[12] Shapiro, A. (2000). The effect of interactive overviews on the development of conceptual structure in novices learning from hypermedia. *Journal of Interactive Multimedia and Hypermedia, 9*(1), 57-78.

[13] Azevedo, R., Guthrie, J.T., & Seibert, D. (2004). The role of self-regulated learning in fostering students' conceptual understanding of complex systems with hypermedia. *Journal of Educational Computing Research, 30*(1), 87-111.

[14] Pintrich, P.R. (2000). The role of goal orientation in self-regulated learning. In M. Boekaerts, P. Pintrich, & M. Zeidner (Eds.), *Handbook of self-regulation* (pp. 451-502). San Diego, CA: Academic Press.

[15] Winne, P.H., & Perry, N.E. (2000). Measuring self-regulated learning. In M. Boekaerts, P. Pintrich, & M. Zeidner (Eds.), *Handbook of self-regulation* (pp. 531-566). San Diego, CA: Academic Press.

[16] Winne, P.H. (2001). Self-regulated learning viewed from models of information processing. In B. Zimmerman & D. Schunk (Eds.), *Self-regulated learning and academic achievement: Theoretical perspectives* (pp. 153-189). Mawah, NJ: Erlbaum.

[17] Zimmerman, B. (2000). Attaining self-regulation: A social cognitive perspective. In M. Boekaerts, P. Pintrich, & M. Zeidner (Eds.), *Handbook of self-regulation* (pp. 13-39). San Diego, CA: Academic Press.

[18] Brusilovsky, P. (2001). Adaptive hypermedia. *User Modeling and User-Adapted Interaction, 11*, 87-110.

[19] Brusilovsky, P. (2004). Adaptive navigation support in educational hypermedia: The role of student knowledge level and the case for meta-adaptation. *British Journal of Educational Technology, 34*(4), 487-497.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Teaching about Dynamic Processes A Teachable Agents Approach

Ruchie Gupta, Yanna Wu, and Gautam Biswas

Dept. of EECS and ISIS, Vanderbilt University
Nashville, TN, 37235, USA
ruchi.gupta, yanna.wu, gautam.biswas@vanderbilt.edu

Abstract. This paper discusses the extensions that we have made to Betty's Brain teachable agent system to help students learn about dynamic processes in a river ecosystem. Students first learn about dynamic behavior in a simulation environment, and then teach Betty by introducing cycles into the concept map representation. Betty's qualitative reasoning mechanisms have been extended so that she can reason about cycles and determine how entities change over time. Preliminary experiments were conducted to study and analyze the usefulness of the simulation. Analysis of the students' protocols was very revealing, and the lessons learnt have led to redesigned simulation interfaces. A new study with the system will be conducted in a fifth grade science classroom in May, 2005.

1. Introduction

Modern society is engulfed by technology artifacts that impact every aspect of daily life. This makes learning and problem solving with advanced math and science concepts important components of the K-12 curriculum. Many of the teaching and testing methods in present day school systems focus on memorization and not on true understanding of domain material [1]. Lack of systematic efforts to demonstrate the students' problem solving skills hamper the students' abilities to retain what they have learned, and to develop the competencies required for advanced science and technology training in the future. A solution proposed by researchers is to introduce constructivist and exploratory learning methods to help students take control of their own learning and overcome the problems of inert learning and learning without understanding [1].

The cognitive science and education literature has shown that teaching others is a powerful way to learn [23]. Preparing to teach others helps one gain a deeper understanding of the subject matter. While teaching, feedback from students provides the teacher with an opportunity to reflect on his or her own understanding of the material [4]. We have adopted the *learning by teaching* paradigm to develop an intelligent learning environment called Betty's Brain, where students teach Betty, a software agent, using a concept map representation [5]. Experiments conducted with Betty's Brain in fifth grade science classrooms demonstrated that the system is successful in helping students learn about river ecosystem entities and their relationships [6]. Students showed improved motivation and put in extra effort to understand the domain material. Transfer tests showed that they were better prepared for "future learning" [7].

In the current version of the system, Betty's representation and reasoning mechanisms are geared towards teaching and learning about *interdependence* in river ecosystems. However, analysis of student answers to post-test questions on *balance* (*equilibrium*) made it clear that students did not quite grasp this concept and how it applied to river ecosystems.

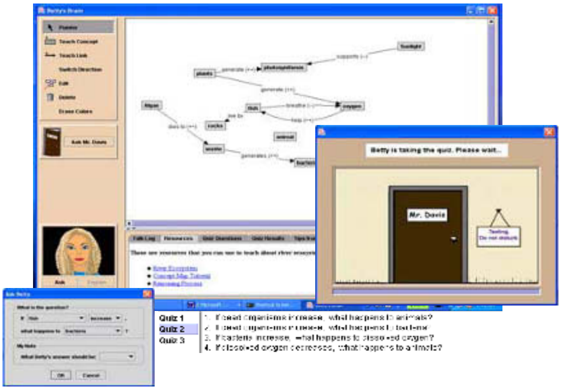
We realized that to understand balance, students had to be introduced to the dynamic behavior of river ecosystems. This brought up two challenges. First, how do we extend students' understanding of interdependence to the notion of balance, and second, how should we extend the representation and reasoning mechanisms in Betty's Brain to help middle school students learn and understand about the behavior of dynamic processes.

Analyzing dynamic systems behavior can be very challenging for middle school students who do not have the relevant mathematical background or maturity. To overcome this, we introduced the notion of cycles in the concept map representation to model changes that happen over time. To scaffold the process of learning about temporal effects, we designed a simulation that provides a virtual window into a river ecosystem in an engaging and easy to grasp manner. This brings up another challenge, i.e., how do we get students to transfer their understanding of the dynamics observed in the simulation to the concept map representation, where changes over time are captured as cyclic structures.

This paper discusses the extensions made to the concept map representation and the reasoning mechanisms that allow Betty to reason with time. A protocol analysis study with high school students pointed out a number of features that we needed to add to the simulation interfaces to help students understand dynamic behaviors. The redesigned simulation interfaces will be used for a study in a middle school science classroom in May 2005.

2. Betty's Brain: Implementation of the Learning by Teaching Paradigm

Betty's Brain is based on the learning by teaching paradigm. Students explicitly teach and receive feedback about how well they have taught Betty. Betty uses a combination of text, speech, and animation to communicate with her student teachers. The teaching process is implemented through three primary modes of interaction between the student and Betty: teach, quiz, query. Fig. 1 illustrates the Betty's Brain system interface. In the teach mode, students teach Betty by constructing a concept map using an intuitive graphical point and click interface. In the query mode, students use a template to generate questions about the concepts they have taught her. Betty uses a qualitative reasoning mechanism to reason with the concept map, and, when asked, she provides a detailed explanation of her answers [5]. In the quiz phase, students can observe how Betty performs on a pre-scripted set of questions. This feedback helps the students estimate how well they have taught Betty, which in turn helps them reflect on how well they have learnt the information themselves. Details of the system architecture and its implementation are discussed elsewhere [589].



1. If dead organisms increase, what happens to animals?
2. If dead organisms increase, what happens to bacteria?
3. If bacteria increase, what happens to dissolved oxygen?
4. If dissolved oxygen decreases, what happens to animals?

Figure 1: Betty's Brain interfaces

The system, implemented as a multi-agent architecture, includes a number of scaffolds to help fifth grade students in science classrooms. These include extensive searchable online resources on river ecosystems and a mentor agent, Mr. Davis, who not only provides feedback to Betty and the student but also provides advice, when asked, on how to be better learners and better teachers. Experimental studies in fifth grade classrooms have demonstrated the success of Betty's Brain in students' preparation for future learning, in general, and learning about river ecosystems, in particular [56].

3. Extending Betty's Brain: Introducing the temporal reasoning framework

One of our primary goals is to help students extend their understanding of interdependence among entities in an ecosystem to the dynamic nature of the interactions between these entities, so that they may reason about and solve problems in real world processes. Middle school students lack the knowledge and maturity to learn about mathematical modeling and analysis approaches for dynamic systems using differential equations. As an alternative, we have to develop intuitive approaches based on simplified, qualitative representations [10, 11] that capture the notion of change over time, hide complex details, but are still accurate enough to replicate the behavior of a real world ecosystem. Even experts use qualitative representations to develop quick, coarse-grained solutions to problems, and explanations for how these solutions are derived [14]. Researchers have used this approach to help students develop high level reasoning skills that are linked to mathematical methods [11].

In this work, we build on the existing qualitative reasoning mechanisms in Betty's Brain to incorporate temporal representations and reasoning. To avoid confusion and cognitive overload, these new additions have to be seamless extensions of the previous representation and reasoning mechanisms. Also, to accommodate our novice student learners, it is important to provide them with scaffolds to aid their understanding of dynamic system behavior. In the learning by teaching framework, the student teachers are given opportunities to learn and understand the material to be taught before they proceed to teach Betty. To help students in their preparations to teach, we have designed and implemented a simulation of a river ecosystem. In the rest of this section, we describe the simulation system, and the extensions to Betty's qualitative reasoning mechanism.

3. 1. The Simulation

In constructivist approaches to learning, students are encouraged to direct and structure their own learning activities to pursue their knowledge building goals [12]. To facilitate this approach to learning, we provide the students with an exploratory simulation environment, where they are exposed to a number of situations that makes them aware of the dynamic phenomena that occur in river ecosystems. The simulation includes a variety of visual tools that the students can use to observe how entities change over time, and how these changes interact to produce cycles of behavior in the ecosystem.

3.1.1 The mathematical model and simulator

The interacting entities in a river ecosystem are typically modeled as differential equation or discrete time state space models. Our river ecosystem simulation is based on a discrete-time model that takes the form: $x_{t+1} = f(x_t, u_t)$, where x_{t+1}, the state vector at time step $t+1$ is defined as a function of the state of the system, x_t, and the input to the system, u_t at time step *t*. We create a one-to-one mapping between the state variables in the simulation, and the entities in the river ecosystem expert concept map that are created by the fifth grade science teachers. This concept map includes the following entities: fish, algae, plants, macro invertebrates, bacteria, oxygen, waste, dead organisms, and nutrients. The quantity of each of these entities is represented by a state variable, and a typical state equation takes on the following form:

$$O_{2_{t+1}} = O_{2_t} + 0.001125.P_t - 0.006.F_t - 0.001.M_t + 0.00075A_t - 0.0004.B_t$$

This linear equation describes the change in the amount of dissolved oxygen, O_2, from one time step to the next for the ecosystem in balance. $O_{2_t}, P_t, F_t, M_t, A_t,$ and B_t represent the quantity of dissolved oxygen, plants, fish, macroinvertebrates, algae, and bacteria, respectively, in appropriate units at time step, *t*. The coefficients in the equation represent the strength of interactions between pairs of entities. For example, the coefficient for F_t is greater than the coefficient for M_t because fish consume more oxygen than macro invertebrates. Producers of oxygen, plants and algae, have positive coefficients and consumers, fish, macroinvertebrates, and bacteria, have negative coefficients in the above equation.

The state equations would have been much more complex with steep nonlinearities, if we had included phenomena, where the river did not remain in balance. Instead, we employ a hybrid modeling approach, and switch the equations when the entities exceed predefined critical values. For example, if the amounts of dissolved oxygen and plants fall below a certain value, they have a strong negative effect on the quantity of fish in the river. This phenomenon is captured by the following equation:

$$\text{If } O2_t < 3\,(\text{ppm}) \text{ and } P_t < 3500\,(\text{micromg/L})$$

$$F_{t+1} = F_t - ((6 - O2_t)/300).F_t - ((4000 - P_t)/50000).F_t$$

Therefore, our state equation-based simulation model captures the behavior of a river ecosystem under different operating conditions that include the behavior of the ecosystem in balance and out of balance.

The simulation model was implemented using AgentSheets [13], which is a software tool designed to facilitate the creation of interactive simulations using a multi agent framework. This tool was chosen primarily because it provides an easy way to construct appealing visual interfaces. Its user friendly drag and drop interface made it easy to implement the simulation model. Each entity was modeled as an agent with the appropriate set of equations describing its behavior at every time step.

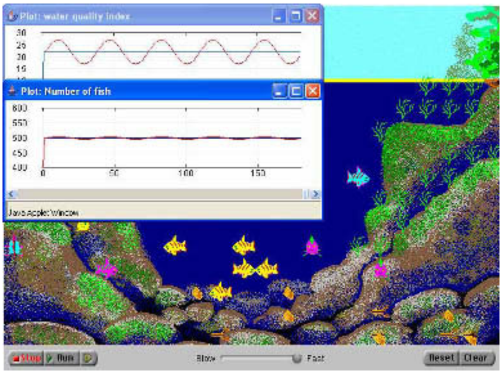


Figure 2: The simulation interface

3.1.2 The visual interface

Fig. 2 illustrates the visual interface of the simulation system. It has two components. The first uses an animation to provide a virtual window into the ecosystem. Its purpose is to give the student an easy to understand global view of the state of the system. The second component uses graphs to give a more precise look at the amount of the different entities and how these amounts change with time. The student can use these graphs to not only determine the amounts, but also study patterns of change. Further, since the cyclic behavior of the variables was clearly visible in these plots, we believed that students could use these graphs to learn about cycle times, and teach Betty this information in the concept map representation.

3.1.3 Ranger Joe

Ranger Joe plays the role of the mentor in the simulation environment. He provides help on a variety of topics that range from textual descriptions of the simulation scenarios, to telling students how to run the simulation, and how to read the graphs. When asked, he makes students aware of the features available in the simulation environment, and how students may use them to learn more about dynamic changes in the river. The current version of Ranger Joe provides responses in text form only.

3.2. Extending Betty's reasoning mechanisms to incorporate temporal reasoning

As discussed earlier, we have extended the concept map representation in Betty's Brain to include cyclic structures. Any path (chain of events) that begins on a concept and comes back to the same concept can be called a cycle. For example, the concepts macroinvertebrates, fish, and dissolved oxygen form a cycle in the concept map illustrated in Fig. 3. Unlike the previous version of Betty's Brain, where the reasoning process only occurred along the paths from the source to the destination concept (identified in the query), e.g., "*If* *fish increase what happens to bacteria*?", the new system also takes into account the changes that occur along feedback paths from the destination to the source concept. For example, a change in the amount of bacteria above may cause a change in the amount of fish along the feedback path, which would further cause a change in bacteria along the forward path and so on. This creates a cycle of change and the time it takes to complete an iteration of the cycle is called the cycle time.

The query mechanism had to be extended so Betty could answer questions that involved change over time, e.g., "*If algae decrease a lot, what will happen to bacteria after one month*?" Last, Betty's reasoning and explanation mechanisms were extended. Each of these is described below.

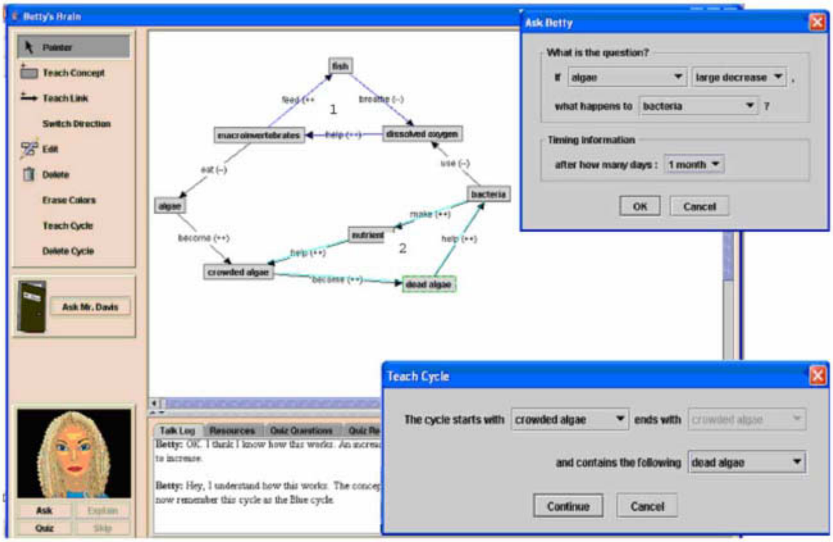
3.2.1. Concept Map Building and Query Interfaces

We extended the concept map interface to allow students to teach Betty about dynamic processes by constructing a concept map with cycles (see Fig. 3). To help Betty identify a cycle in the concept map, students click on the "Teach Cycle" button, which brings up a pop up window with the same name. Students identify the cycle, using any one of the nodes as the starting point, e.g., *crowded algae* in cycle 2 (Fig. 3) then identify the other concepts in the cycle in sequence, e.g., *dead algae*, then *bacteria*, and then *nutrients*. Along with each cycle, the student also has to teach Betty the time (in days) it takes to complete an iteration of the cycle. Betty responds by identifying the cycle with a number. Fig. 3 shows the concept map after the student has built two cycles identified by Betty as cycles 1 and 2 with cycle times of 5 and 10 days, respectively.

Like before, students can query Betty. The original query templates were extended as shown in Fig. 3 to include a time component.

3.2.2. Temporal Reasoning Algorithm and Explanation Process

The extended temporal reasoning algorithm that Betty uses has four primary steps. In step 1, Betty identifies all the forward and feedback paths between the source and destination concepts in the query. For the query, "*If algae decrease a lot, what will happen to bacteria after one month*?" Betty identifies *algae* as the source concept and *bacteria* as the destination concept. A forward path is a path from the source to the destination concept (e.g., *algae* → *crowded algae* → *dead algae* → *bacteria*) and the feedback path traces back from



What is the question?

If algae large decrease, what happens to bacteria ?

Timing information

after how many days : 1 month

The cycle starts with crowded algae ends with crowded algae

and contains the following dead algae

Betty: OK. I think I know how this works. An increa
to increase.

Betty: Hey, I understand how this works. The concep
now remember this cycle as the Blue cycle.

Figure 3: Betty's Brain: Temporal Reasoning Interface

(top-right): temporal question template; (bottom-right): interface for teaching Betty about cy-

the destination to the source concept (e.g., *bacteria* → *dissolved oxygen* → *macroinvertebrates* → *algae*). In step 2, using the original reasoning process [5], all the concepts on these paths are given an initial value. In step 3, Betty orders the cycles from slowest to fastest, and executes the propagation of the chain of events for each cycle. When a path includes more than one cycle, the faster cycle is run multiple times, and then its effects are integrated with the chain of events propagation in the slower cycle. This method incorporates the time-scale abstraction process developed by Kuipers [014]. This process is repeated for the feedback path, and the result gives the updated values for the source and destination concepts after one full cycle. In step 4, this process is repeated multiple times till the value of the destination concept has been derived for the time period stated in the query.

For example, when asked the query about algae and bacteria, Betty first identifies the forward and feedback paths shown earlier, and propagates the change of algae to the concepts on the forward path and then to the concepts on the feedback path using the original reasoning mechanism. She determines that *crowded algae*, *dead algae* and *bacteria* decrease a lot on the forward path, and *dissolved oxygen*, and *macroinverterbrates* increase a lot. In step 2, she identifies two cycles (cycles 1 and 2 in Fig. 3), one on the forward path, and the second on the feedback path. Since cycle 2 has the larger cycle time, she assigns the main cycle a period of 10 days. After that, she runs the reasoning process twice (10/5) for cycle 1 and determines that *macroinverterbrates* and *fish* increase a lot and *dissolved oxygen* decreases a lot. Cycle 2 is run once (10/10) to derive that *crowded algae, dead algae,* and *nutrients* decrease a lot. Betty then combines the effects of cycles 1 and 2 to determine the value for *algae* after 10 days (feedback effect), i.e., *algae* decrease a lot, and, as a result, *bacteria* decrease a lot (this completes one cycle, i.e., a 10 day period of behavior). Since the student wanted to know what happens to *bacteria* after one month, this process has to be repeated three times, and Betty arrives at the answer that *bacteria* decrease a lot.

To facilitate student's understanding of the temporal reasoning mechanisms, Betty uses a top-down explanation process, if asked to explain her answer. First, Betty explicates her final answer, and states how many full cycles she had to run to get this answer. Then Betty breaks down the rest of the explanation cycle by cycle, and then combines the results. Students can control what parts of the explanation and how much detail they want, by simply clicking on "Continue Explanation," "Repeat," and "Skip" buttons in left bottom of the interface.

4.0 Protocol Analysis Studies with the Temporal Betty

We conducted a preliminary protocol analysis study with 10 high school students. None of these students knew or remembered much about the river ecosystems unit they had covered in middle school. The overall goal for each student was to teach Betty about the dynamic processes in river ecosystems by first teaching her about general concepts of the ecosystem by drawing a concept map and then refining the map by identifying cycles and teaching her timing information. One of our goals was to see how they would use the simulation tool to derive information about the structure and time period of cycles. Each student worked with a research assistant (who conducted the study) on the Betty's Brain system for two one hour sessions. As students worked, the research assistants involved them in a dialog, in which they asked the students to interpret what they saw in the simulation, and how that information may be used to teach Betty using the concept map structure. All verbal interactions between the student and the researcher was taped, and later transcribed and analyzed. An overview of the results is presented next.

Overall, all students liked the simulation and felt that it was a good tool for learning about river ecosystems. Also, they thought that the river animation was engaging and served the purpose of holding the student's attention. The researchers asked specific questions that focused on students' understanding of graphs, cycles and cycle times. An example dialog that was quite revealing is presented below.

Researcher: So do you think the graphs were helpful in helping you think about the temporal cycles?
Student: They were critical because that's where I got my initial impression because ordinarily when someone gives you something to read, it's really a small amount of text and doesn't clarify much. So the graphs are the main source of information.

Also, some of the dialogues indicated that the graphs were put to good use in learning about cycle times. For example, a student, who was trying to find the cycle time involving fish and macro invertebrates said:

Researcher: Are you trying to assign the time period of the cycle?
Student: Yeah, see how the cycle kind of completes the whole graph in about 2 days.

A second example:

Researcher: What is hard about using the graphs?
Student: Well, I see the graph; I see the sine wave and the period of the sine wave, right here, right?
Researcher: Right.
Student: So I would think of that as completing the cycle.

Students also made some important suggestions about the graphs. Many of them mentioned that it would be better to have multiple quantities plotted on the same graph. Some of them said that it would be useful to have quantities plotted against each other rather than plotted against time so that relationships between such quantities could be observed directly. Others said that simply showing numbers of changing quantities over time would be useful too.

We also had some feedback about the resources and feedback that Ranger Joe provided. The students found the text resources to be useful but thought there was too much to read, so it would be a good idea to reorganize the text into sections and make it searchable. They also thought that Ranger Joe was passive, and that he should be an active participant in the learning process. Most students stressed the importance of being able to easily navigate between different graphs and see them side by side for easy comparisons.

These protocols provided valuable feedback on the effectiveness of the different features of the simulation. We realized some of the features would have to be modified, and extra features had to be implemented. These changes could not be implemented in AgentSheets. This motivated us to redesign and reimplement the simulation in a flexible programming environment like Java to facilitate the addition of new tools and easy integration of the simulation and Ranger Joe with the temporal Betty system.

5.0 The Redesigned Simulation System

Different representations enhance different aspects of thinking and problem solving skills. In the new simulation, we present the state of the river ecosystem using a number of different representations that are more relevant to their problem-solving tasks. In this version of the simulation, we provide the students with a new set of tools which exploits the use of representations as a critical tool of thought. We also hope that this will help students develop representational fluency, which is an important attribute to have while attempting to solve complex real world problems.

The tools for the presentation and analysis of the information in the graphs have been revamped. Students can now choose the graphs they want to view from a pull-down menu. They can choose between line graphs and bar graphs. The unit of time for the bar graph plots can be a day (unit of time in the simulation), or a week (typically the frequency with which experimental data is collected in rivers). A second feature introduced is a compare graph tool that allows the student to plot multiple quantities in the same graph to get a better idea of the interrelationships between the entities. The students can also view the simulation data in tabular form. A third tool will help students analyze the temporal change in the quantities in a more abstract qualitative way. Changing trends are depicted by upward facing arrows (increase in the quantity) and downward facing arrows (decrease in the quantity). This representation provides information that is closer to what students need to generate the concept map.

The text resources have been restructured and reorganized in a hypertext form. They contain a detailed description of how to use the different tools in the simulation and how to use and interpret graphs. A keyword search features helps students to easily find the specific information they are looking for. The mentor agent, Ranger Joe, plays a more active role in this new environment. He can address specific questions that the student might have, and gives feedback that is tailored to the students' current activities.

5.0 Discussion and Future Work

Our upcoming study with middle school students starting in May, 2005 will focus on evaluating the usefulness of the system (temporal Betty + the simulation) in teaching about dynamic processes in a river ecosystem. In particular, we want to find how easy it is for students to understand the notion of timing and cycles and also how well they can learn to translate timing information in the simulation into the concept map framework. Also, we want to study the various graph representations in terms of their general usefulness, their frequency of use, and their success in helping students learn about the dynamic nature of ecosystem processes.

Acknowledgements: This project is supported by NSF REC grant # 0231771.

References

1. Bransford, J.D., A.L. Brown, and R. R. Cocking (2001). How People Learn: Brain, Mind, Experience and School.
2. Palinscar, A. S. & Brown, A. L. (1984). Reciprocal teaching of comprehension-fostering and comprehension -monitoring activities. Cognition and instruction, 1: 117-175.
3. Bargh, J. A., & Schul, Y. (1980). On the cognitive benefits of teaching. *Journal of Educational Psychology, 72*(5), 593-604
4. Webb, N. M. (1983). Predicting learning from student interaction: Defining the interaction variables. *Educational Psychologist*, 18, 33-41.
5. Biswas, G., D. Schwartz, K. Leelawong, N. Vye, and TAG-V (2005). "Learning by Teaching: A New Agent Paradigm for Educational Software," *Applied Artificial Intelligence*, special issue on Educational Agents, 19(3): 363-392.
6. Biswas, G., Leelawong, K., Belynne, K., et al. (2004). Incorporating Self Regulated Learning Techniques into Learning by Teaching Environments. *26th Annual Meeting of the Cognitive Science Society*, (Chicago, Illinois, 120-125.
7. Schwartz, D. L. and Martin, T. (2004). Inventing to prepare for learning: The hidden efficiency of original student production in statistics instruction. *Cognition & Instruction,* 22: 129-184.
8. Biswas, G., Leelawong, K., Belynne, K., et al. (2004). Developing Learning by Teaching Environments that support Self-Regulated Learning. in *The seventh International Conference on Intelligent Tutoring Systems*, Maceió, Brazil, 730-740.
9. Leelawong, K., K. Viswanath, J. Davis, G. Biswas, N. J. Vye, K. Belynne and J. B. Bransford (2003). Teachable Agents: Learning by Teaching Environments for Science Domains. *The Fifteenth Annual Conference on Innovative Applications of Artificial Intelligence*, Acapulco, Mexico, 109-116.
10. Bredeweg, B., Struss, P. (2003). Current Topics in Qualitative Reasoning (editorial introduction). *AI Magazine*, 24(4), 13-16.
11. Bredeweg, B., Forbus, K. (2003). Qualitative Modeling in Education. *AI Magazine*, 24(4). 35-46.
12. Harel, I., and Papert, S. (1991). Constructionism. Norwood, NJ: Ablex.
13. Repenning, A. and Ioannidou (2004). Agent-Based End-User Development. Communications of the ACM, 47(9), 43-46.
14. Kuipers, B. (1986). Qualitative Simulation, Artificial Intelligence, 29: 289-388.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Exam Question Recommender System

Hicham HAGE   Esma AÏMEUR

Department of Computer Science and Operational Research
University of Montreal
{hagehich, aimeur}@iro.umontreal.ca

Abstract. Although E-learning has advanced considerably in the last decade, some of its aspects, such as E-testing, are still in the development phase. Authoring tools and test banks for E-tests are becoming an integral and indispensable part of E-learning platforms, and with the implementation of E-learning standards, such as IMS QTI, E-testing material can be easily shared and reused across various platforms. With this extensive E-testing material and knowledge comes a new challenge: searching for and selecting the most adequate information. In this paper we propose using recommendation techniques to help a teacher search for and select questions from a shared and centralized IMS QTI-compliant question bank. Our solution, the Exam Question Recommender System, uses a hybrid, feature-augmentation, recommendation approach. The recommender system uses Content-Based and Knowledge-Based recommendation techniques, resorting to the use of a new *heuristic function*. The system also engages in collecting both implicit and explicit feedback from the user in order to improve on future recommendations.

Keywords: E-learning, E-testing, Assessment tool, E-learning Standards, IMS QTI, Hybrid Recommendation.

1 Introduction

E-learning has advanced considerably in the last years. Today, there exist many E-learning platforms, commercial (*WebCT* [11], *Blackboard* [10]) or open source (*ATutor* [19]), which offer many tools and functionalities [16]. Some of these tools are aimed towards teachers and developers, and other tools aimed towards students and learners [5]. Although E-learning has come a long way, some of its aspects are still in their early stages. One such aspect is E-testing. While existing E-learning platforms do offer E-testing Authoring tools, most are only basic E-testing functionalities [16] [18], which are limited to the platform itself. With the emergence of E-learning standards and specifications, such as the *IMS QTI* [14] (IMS Question and Test Interoperability), E-learning material can be reusable, accessible, interoperable, and durable. With E-learning standards, E-testing material can be transferred from one platform to another. Furthermore, some E-learning platforms are starting to offer the functionality of Test Banks. This feature allows teachers and developers to save their questions and exams in the Test Bank for future access and use. To the best of our knowledge, E-learning platforms Test Banks are *limited* to the teacher's private use, where each teacher can only access his personal, private questions and tests. Therefore, in order to share available E-testing knowledge, teachers must do so explicitly by using import/export functionalities offered only by some platforms. Consequently, due to the limitations in knowledge sharing, the size of the Test Banks remains relatively small, thus E-learning platforms only offer *basic filters* to search for information within the Test Bank. In order to encourage knowledge sharing and reuse, we are currently in the works of implementing a web-based *assessment authoring tool* called Cadmus. Cadmus offers an IMS QTI-compliant centralized questions- and-exams repository for teachers to store and share E-testing knowledge and resources. For such a repository to be beneficial it must contain extensive information on questions and exams. The bigger and more useful the repository becomes, the more dreadful is the task to search for and retrieve necessary information and material. Although there exist tools to help teachers locate learning material [8] [9], to our knowledge there aren't *personalized* tools to help the teacher select exam material from a shared data bank. What we propose is to incorporate into Cadmus an *Exam Question Recommender System* to help teachers find and select questions for exams. The recommender uses a hybrid, feature-augmentation recommendation approach. The first level is a *Content-Based* filter, and the second level is a *Knowledge-Based* filter [2] [3]. In order to recommend questions, the Knowledge-Based filter resorts to a *heuristic function*. Furthermore, the Exam Question Recommender System gathers *implicit* and *explicit feedback* [4] from the user in order to improve future recommendations.

The paper is organized as follows: section 2 introduces E-learning, E-testing, and offers an overview of E-learning standards, in particular IMS QTI; section 3 presents current recommendation techniques; section 4 describes the architecture and approach of the Exam Question Recommender System; section 5 highlights the testing procedure and the results; and section 6 concludes the paper and presents the future works.

2 E-learning

E-learning can be defined with the following statement: the delivery and support of educational and training material using computers.

E-learning is an aspect of distant learning, where teaching material is accessed through electronic media (internet, intranet, CD-ROM …) and where teachers and students can communicate electronically (email, chat rooms ...). E-learning is very convenient and portable. Furthermore, E-learning involves great collaboration and interaction between students and tutors or specialists. Such collaboration is made easier by the online environment. For example, a student in Canada can have access to a specialist in Europe or

Asia through email or can assist in the specialist's lecture through a web conference. There are four parts in the life cycle of E-learning [17]: *Skill Analysis*, *Material Development*, *Learning Activity* and *Evaluation/Assessment*.

2.1 E-testing

There exist many E-learning platforms, such as Blackboard, WebCT and ATutor that offer different functionalities [16]. Although Evaluation and Assessment is an important part of the E-learning life cycle, E-testing remains in its early development stages. Most E-learning platforms do offer E-testing Authoring tools, most of which offer only basic testing functionalities, and are limited to the platform itself. For instance, most E-learning platforms offer support for basic question types such as Multiple Choice, True/False and Open-Ended Questions, but do not offer the possibility of adding multimedia content (images, sounds …), to set a time frame for the exam, or even include import functionalities to add questions from external sources [16]. In order to deliver E-learning material, each E-learning platform chooses different delivery media, a different platform/operating system and its own unique authoring tools, and stores the information in its own format. Therefore, in order to reuse E-learning material developed on a specific platform, one must change considerably that material or recreate it using the target platform authoring tools—hence increasing the cost of development of E-learning material. Standards and specifications help simplify the development, use and reuse of E-learning material.

2.2 IMS Question and Test Interoperability

As stated in the ADL (Advanced Distributed Learning) goals [13], standards and specifications ensure that E-learning material is: *Reusable* (modified easily and usable on different development tools), *Accessible* (available as needed by learners or course developers), *Interoperable* (functional across different hardware or software platforms), and *Durable* (easy to modify and update for new software versions). Currently, there are many organizations developing different standards for E-learning [15], each promoting its own standards. Some of the leading organizations with the most widely accepted standards are: *IEEE Learning Technology Standards Committee* [12], *ADL Initiative (Advanced Distributed Learning)* [13], and *IMS Project (Instructional Management System)* [14]. IMS QTI sets a list of specifications used to exchange assessment information such as questions, tests, and results. QTI allows assessment systems to store their data in their own format, and provides a means to import and export that data in the QTI format between various assessment systems.

With the emergence and use of E-learning standards, learning and testing material can be reused and shared among various E-learning platforms [7]. Knowledge sharing would lead to a quick increase in the available information and material, leading to the need for recommendation systems to help filter the required data.

3 Recommender System

Recommender systems offer the user an automated recommendation from a large information space [6]. There exist many recommendation techniques, differentiated upon the basis of their knowledge sources used to make a recommendation. Several recommendation techniques are identified in [2] including: *Collaborative Recommendation* (the recommender system accumulates user ratings of items, identifies users with common ratings, and offers recommendations based on inter-user comparison), *Content-Based Recommendation* (the recommender system uses the features of the items, and the user's interest in these features to make a recommendation), and *Knowledge-Based Recommendation* (the recommender system bases the recommendation of items on inferences about the user's preferences and needs). Each recommendation technique has its advantages and limitations, thus the use of hybrid systems that combines multiple techniques to produce the recommendation. There exist several techniques of hybridization [1] [2] such as: *Switching* (the recommender system switches between several techniques, depending on the situation, to produce the recommendation), *Cascade* (the recommender system uses one technique to generate a recommendation, and a second technique to break any ties), and *Feature Augmentation* (the recommender system uses one technique to generate an output, which in turn is used as input to a second recommendation technique). Our Exam Question Recommendation System uses a hybrid, feature-augmentation approach, using Content-Based and Knowledge-Based recommendation.

4 Exam Questions Recommendation System Architecture

Cadmus is an E-testing platform that offers teachers an extensive question library. The more comprehensive Cadmus's question library is, the harder the task to search for and select questions. The first suggestion that comes to mind is to filter questions according to their content and the needs of the teacher. A Content-Based filter will help, but might not be enough. For instance, there might be between 50 and 100 questions in the library that satisfy the content requirement, but not all will be rated the same by different teachers with different preferences: a teacher might prefer "multiple choice" to "true and false", or might prefer questions with a certain level of difficulty. What we propose is a *feature-augmentation*, *hybrid-recommendation* approach, where the first level is a Content-Based filter and the second level a Knowledge-Based filter. The Content-Based filter will reduce the search to questions with content pertinent to the teacher's needs, and the Knowledge-Based filter will sort these questions with regards to the teacher's preferences, such that the higher ranking questions are the most likely to be chosen by the teacher. Figure 1 illustrates the architecture of the recommender system. We can distinguish two different types of components: *Storage components* (Question Base and User Profile) and *Process Components* (Content-Based Filter, Knowledge-Based Filter and Feedback).

4.1 Question Base

The *Question Base* stores all the questions created by the teachers. The actual question is stored in an external XML file following the IMS QTI specifications, and the database contains the following information about the question:

- **Ident**: unique question identifier
- **Title**: contains the title of the question
- **Language**: corresponds to the language of the question, i.e. English, French …
- **Topic**: denotes the topic of the question, i.e.: Computer Science, History…
- **Subject**: specifies the subject within the topic, i.e.: Databases, Data Structures …
- **Type**: denotes the type of question, i.e.: multiple choice, true/false …
- **Difficulty**: specifies the difficulty level of the question, according to possible values: Very Easy, Easy, Intermediate, Difficult, and Very Difficult
- **Keywords**: contains keywords relevant to the question's content
- **Objective**: corresponds to the pedagogical objective of the question: Concept Definition, Concept Application, Concept Generalization, and Concept Mastery
- **Occurrence**: a counter of the number of exams this question appears in
- **Author**: the author of the question
- **Availability**: designates whether the question is available only to the author, to other teachers, or anyone
- **QTIQuestion**: handle to the IMS QTI-compliant XML file where the question and all of relevant information such as answers, comments, and hints are stored

4.2 User Profile

The User Profile stores information and data about the teacher that are used by the Knowledge-Based filter. The user profile contains the following:

- **Login**: unique identifier of the user
- **Type Weight**: selected by the user for the type criteria
- **Occurrence Weight**: specified by the user for the occurrence criteria
- **Difficulty Weight**: chosen by the user for the difficulty criteria
- **Author Weight**: specified by the user for the author criteria
- **Individual Type Weights**: system-calculated weight for each different question type, i.e. weight for True/False, for Multiple Selection …
- **Individual Occurrences Weights**: system-calculated weight for each different question occurrence, i.e. Very Low, Average, High …
- **Individual Difficulties Weights**: system-calculated weight for each different question difficulty, i.e. weight for Easy, for Difficult…
- **Individual Authors Weights**: system-calculated weight for each author

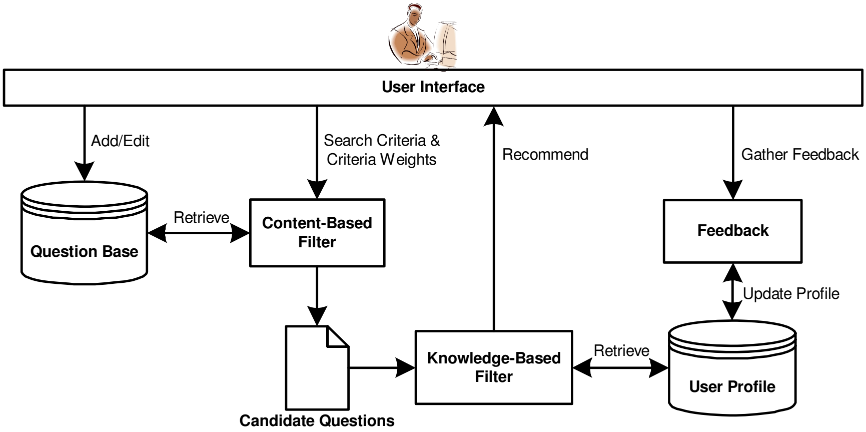


Figure 1: System Architecture

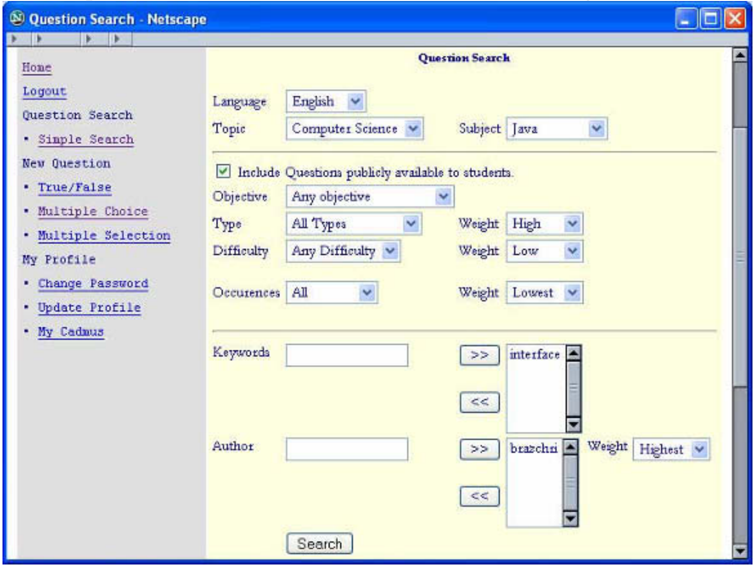
The teacher-specified Type, Occurrence, Difficulty, and Author weights are set manually by the teacher. These weights represent his criteria preference, i.e. which of the four independent criteria is more important for him. The teacher can select one out of five different values with each assigned a numerical value (Table 1) that is used in the distance function explained in 4.4.1. The system-calculated weights infer the teacher's preferences of the various values each criteria might have. For example, the Type criteria might have one of three different values: True/False (TF), Multiple Choice (MC) or Multiple Selection (MS), thus the system will calculate three different weights: w_{TF}, w_{MC} and w_{MS}. The system keeps track of a counter for each individual weight (i.e. a counter for True/False, a counter for Multiple Selection …), and a counter for the total number of questions selected thus far by the teacher. Each time the teacher selects a new question, the counter for the total number of questions is incremented, and the corresponding individual weight is incremented accordingly, i.e. if the question is a True/False, then the True/False counter is incremented, and w_{TF} = Counter (True/False) / Total number of questions. The value of the individual weights is the percentage of usage, so that if the user selected 100 questions out of which 33 were TF, 59 were MC, and 8 were MS, then $w_{TF} = 0.33$, $w_{MC} = 0.59$, $w_{MS} = 0.08$, and $w_{TF} + w_{MC} + w_{MS} = 1$.

Table 1: Weights Values

Weight	*Lowest*	*Low*	*Normal*	*High*	*Highest*
Value	0.25	0.5	1	2	4

4.3 Content-Based Filter

When, for the purpose of creating a new exam, the teacher wants to search for questions, he must specify the search criteria for the questions (Figure 2). The search criteria are used by the Content-Based Filter and consist of the following: *Language*, *Topic*, *Subject*, the option of whether or not to include *questions* that are *publicly available to students*, *Objective*, *Type*, *Type Weight* (used by the teacher to specify how important this criteria is to him, compared with other criteria), *Difficulty*, *Difficulty Weight*, *Occurrence*, *Occurrence Weight*, *Keywords* (only the questions with one or more of the specified keywords are retrieved. If left blank, the question's keywords are ignored in the search), *Author* (only the questions of the specified author(s) are retrieved), and *Author Weight*.



Question Search

Language: English
Topic: Computer Science Subject: Java

☑ Include Questions publicly available to students.
Objective: Any objective
Type: All Types Weight: High
Difficulty: Any Difficulty Weight: Low
Occurences: All Weight: Lowest

Keywords: interface
Author: brazchri Weight: Highest

Figure 2: Question Search

The teacher must first select the *language* and the *topic* for the question, and has the option to restrict the search to a specific *subject* within the selected topic. Since some questions may be *available to students*, the teacher has the *option to include or omit* these questions from the search. Furthermore, the teacher may restrict the search to a certain question *objective*, question *type*, question *occurrence*, and question *difficulty*. Moreover, the teacher can narrow the search to questions from one or more *authors*, and can refine his search further by specifying one or more *keywords* that are relevant to the question's content. Finally, the teacher can specify the weight, or the importance of specific criteria (this weight is used by the Knowledge-Based filter). When the user initiates the search, the recommender system will start by collecting the search criteria and weights. Then the search criteria are constructed into an SQL query that is passed to the database. The result of the query is a collection of *candidate questions* whose content is relevant to the teacher's search. The candidate questions and the criteria weights are then used as the input to the Knowledge-Based filter.

4.4 Knowledge-Based Filter

The Knowledge-Based Filter takes as input the candidate questions and the criteria weights. The criteria weight is specified by the teacher, and represents the importance of this specific criteria to the user compared to other criteria. Table 1 presents the possible values of the criteria weight and the respective numerical values. The Knowledge-Based filter retrieves the teacher's profile from the User Profile repository, and uses the distance function to calculate the distance between each of the candidate questions and the teacher's preferences.

4.4.1 Distance Function

In order to decide which question the teacher will prefer the most; we need to compare several criteria that are unrelated. For instance, how can someone compare the Type of a question with the number of times it appears in exams (the Occurrence)? Since we cannot correlate the different criteria, we left this decision to the teacher: he must select the criteria weight. This weight must either reinforce or undermine the value of the criteria. The Knowledge-Based recommender uses a heuristic Distance Function (Equation 1) to calculate the distance between a question and the teacher's preferences.

$$s = \sum_i W_i w_j$$

Equation 1: Distance Function

The distance function is the sum of the products of two weights, W and w, where W is the weight specified by the teacher for the criteria and w is the weight calculated by the recommender system. The multiplication by W will either reinforce or undermine the weight of the criteria. Consider the following example to illustrate the distance function: in the search performed in Figure 2, the teacher set W_{Type} = High, $W_{\text{Difficulty}}$ = Low, $W_{\text{Occurence}}$ = Lowest and W_{Author} = Highest (values illustrated in Table 1). Table 2 illustrates the values of two different questions, and Table 3 illustrates the individual weights retrieved from the teacher's profile. Table 3 contains only a part of the actual profile, reflecting the data pertinent to the example.

Table 2: Question Values

	Type	*Difficulty*	*Occurrence*	**Author**
Question1 (Q1)	True/False	Easy	High	Brazchri
Question2 (Q2)	Multiple Choice	Easy	Low	Brazchri

Table 3: Teacher's Profile Values

Criteria	*Type*		*Difficulty*	*Occurrence*		*Author*
Value	True/False	Multiple Choice	Easy	High	Low	Brazchri
Weight	0.33	0.11	0.5	0.06	0.54	0.15

Calculating the distance function for both questions will give:

(1)
$$s_1 = (W_{Type} \times w_{True/False}) + (W_{Difficulty} \times w_{Easy}) + (W_{Occurence} \times w_{High}) + (W_{Author} \times w_{Hala})$$
$$s_2 = (W_{Type} \times w_{MultipleChoice}) + (W_{Difficulty} \times w_{Easy}) + (W_{Occurence} \times w_{Low}) + (W_{Author} \times w_{Hala})$$

(2)
$$s_1 = (2 \times 0.33) + (0.5 \times 0.5) + (0.25 \times 0.06) + (4 \times 0.15) = 1.525$$
$$s_2 = (2 \times 0.11) + (0.5 \times 0.5) + (0.25 \times 0.54) + (4 \times 0.15) = 1.25$$

Although there exists a big difference between the Occurrences' weights in the favor of Q2, Q1 will rank higher because the teacher deemed the Type criteria as more important than the Occurrence criteria.

4.5 Feedback

The Exam Question Recommender System first retrieves candidate questions using the Content-Based filter, then ranks the candidate questions using the Knowledge-Based filter, and finally displays the questions for the teacher to select from. The teacher can then select and add the desired questions to the exam. At this stage the exam creation and its effect on the questions and teacher's profile is only simulated; no actual exam is created. The Exam Question Recommender System gathers the feedback from the teacher in two manners: *Explicit* and *Implicit*. Explicit feedback is gathered when the teacher manually changes the criteria weights, and his profile is updated with the new selected weight. Implicit feedback is gathered when the teacher selects and adds questions to the exam. Information such as the question type, difficulty, occurrence and author is gathered to update the *system-calculated* individual weights in the teacher's profile (as highlighted in 4.2).

5 Testing and Results

The purpose of the Exam Question Recommender System is to simplify the task of searching for and selecting questions for exams. The aim of the testing is to determine the performance of the recommendation in helping the teacher select questions. To test the recommender system, we used a database containing about 200 Java questions. The system has a total of 33 different authors/users. For each recommendation and selection, the system recorded the following: *Teacher's Name*, *Date*, *Search Number*, *Questions Recommended*, *Questions Selected*, and *Rank*. The date and the search number enable us to track the performance and quality of the recommendation as the user makes more choices and his profile is developing. The rank of the selected questions is an indication of the accuracy of the Knowledge-Based Filter, the higher the rank of the selected questions, the more accurate is the recommendation of the Knowledge-Based filter.

5.1 Results

The preliminary results are very encouraging and we are still undergoing further testing. There were 33 registered users (teachers, teacher's assistants and graduate students) testing the system for a total of 89 recommendations, and 366 questions selected and added to exams (some questions were selected more than once). On average 40 questions were recommended after each search. Figure 3 illustrates the Ranking Partition of the selected questions. Almost 55% of the selected questions were among the top ten recommended questions. Figure 4 illustrates the rank partitioning of the questions selected among the top 10. We notice that the first ranking question is the most selected, while the top five ranked questions constitute about 75% of the selected questions within the top ten ranked by the recommender system. On an average of 40 questions proposed with each search, almost 55% of the selected questions were within the first ten questions recommended by the Exam Question Recommender System, and almost 75% were within the first 20 recommended questions. Thus far, we can conclude that in 75% of the cases, the teacher did not need to browse farther than 20 questions, thereby making it easier for the teacher to search for the required questions for his exam.

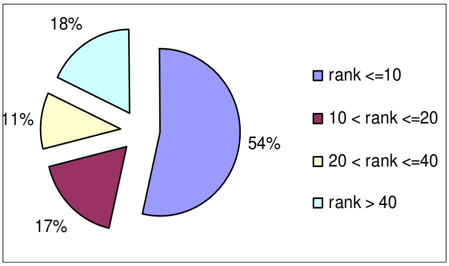


Figure 3: Ranking Partition

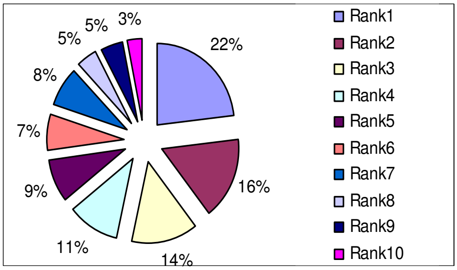


Figure 4: Top Ten Ranking Partition

6 Conclusion

Today, many E-learning platforms offer authoring tools for E-testing. These authoring tools create E-testing material that will remain mostly confined to the teacher and the platform itself. We are in the process of creating an alternative solution, Cadmus, which offers an independent, IMS QTI-compliant platform to create and share E-testing material. Compared to other platform's (WebCT, Blackboard, and ATutor) E-testing authoring tools, Cadmus has the advantage of simplifying knowledge sharing. Teachers can choose which material to share, and with whom (Teachers or Students). In addition, since Cadmus stores the questions and exams following the IMS QTI specifications, E-testing material within Cadmus can be easily shared to other E-learning platforms that offer support to IMS QTI and import/export functionality. Furthermore, to help the teachers in their search for information, we proposed the Exam Question Recommender System, which has been tested on a Question Bank of around 200 questions with 33 different users. Preliminary results have shown that the recommendation of the questions is worthwhile. On an average of 40 questions proposed at each search, almost 55% of the selected questions were within the first ten questions recommended by the Exam Question Recommender System, and almost 75% of the selected questions were within the first 20 recommended questions.

What we propose next is to take the Exam Question Recommender System a step further. We propose to enrich the Teacher's profile to include more information associating the various search criteria. For example, a certain Teacher might associate True/False questions as being "Easy" questions, or prefer the Multiple Selection questions of one author and the True/False questions of another. In addition, by including in the Teacher's profile information about his approach, methodology, and exam structure, we can create personalized exam templates and then use the Exam Question Recommender System to fill these templates with questions and help automate the exam creation process.

References

[1] Burke, R.: "Hybrid Recommender Systems with Case-Based Components". *Advances in Case-Based Reasoning, 7th European Conference (ECCBR 2004)*, pp 91-105, Madrid, 2004.

[2] Burke, R.: "Hybrid Recommender Systems: Survey and Experiments". *User Modeling and User-Adapted Interaction*, Vol. 12, No. 4, pp 331-370, 2002.

[3] Breadely, K., and Smyth, B.: "An Architecture for Case-Based Personalized Search". *Advances in Case-Based Reasoning, 7th European Conference (ECCBR 2004)*, pp 518-532, Madrid, 2004.

[4] Cabena, P., Hadjinian, P., Stadler, R., Verhees, J., and Znasi A.: "Discovering Data Mining: From Concept To Implementation". Upper Saddle River, NJ: Prentice-Hall, 1997.

[5] Gaudiosi, E., Boticario, J.: "Towards web-based adaptive learning community". *International Conference on Artificial Intelligence in Education (AIED 2003)*, pp 237-244, Sydney, 2003.

[6] Miller, B., Konstan, J., and Riedl, J.: "PocketLens: Toward a Personal Recommender System". *ACM Transaction on Information Systems*, Vol. 22, No. 3, pp 437-476, 2004.

[7] Mohan, P., Greer, J.: "E-learning Specification in the context of Instructional Planning". *International Conference on Artificial Intelligence in Education (AIED 2003)*, pp 307-314, Sydney, 2003.

[8] Tang, T., McCalla G.: "Smart Recommendation for an Evolving E-Learning System". *International Conference on Artificial Intelligence in Education (AIED 2003)*, pp 699-710, Sydney, 2003.

[9] Walker, A., Recker, M., Lawless, K., & Wiley, D.: "Collaborative information filtering: A review and an educational application". *International Journal of Artificial Intelligence and Education*, Vol. 14, pp 3-28, 2004.

[10] http://blackboard.com/

[11] http://www.webct.com

[12] http://ltsc.ieee.org/

[13] http://www.adlnet.org/

[14] http://www.imsproject.org/

[15] http://workshops.eduworks.com/standards/

[16] http://www.edutools.info/index.jsp

[17] http://www.asia-elearning.net/content/aboutEL/index.html

[18] http://www.marshall.edu/it/cit/webct/compare/comparison.html#develop

[19] http://www.atutor.ca/

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

DIANE, a diagnosis system for arithmetical problem solving

Khider Hakem*, Emmanuel Sander*, Jean-Marc Labat**, Jean-François Richard*
** Cognition & Usages, 2 Rue de la Liberté, 93526 Saint Denis Cedex 02France*
khider.hakem@cognition-usages.org, sander@univ-paris8.fr, richard@univ-paris8.fr
*** UTES, Université Pierre et Marie Curie, 12, rue Cuvier 75270 Paris cedex 05, France*
jean-marc.labat@upmc.fr

Abstract. We hereby describe DIANE an environment that aims at performing an automatic diagnosis on arithmetic problems depending on the productions of the learners. This work relies on results from cognitive psychology studies that insist on the fact that problem solving depends to a great extent on the construction of an adequate representation of the problem, which is highly constrained. DIANE allows large-scale experimentations and has the specificity of providing diagnosis at a very detailed level of precision, whether it concerns adequate or erroneous strategies, allowing one to analyze cognitive mechanisms involved in the solving process. The quality of the diagnosis module has been assessed and, concerning non verbal cues, 93.4% of the protocols were diagnosed in the same way as with manual analysis.
Key Words: cognitive diagnosis, arithmetical problem solving, models of learners.

Introduction

DIANE (French acronym for Computerized Diagnosis on Arithmetic at Elementary School) is part of a project named « conceptualization and semantic properties of situations in arithmetical problem solving » [12]; it is articulated around the idea that traditional approaches in terms of typologies, schemas or situation models, the relevance of which remains undisputable, do not account for some of the determinants of problem difficulties: transverse semantic dimensions, which rely on the nature of the variables or the entities involved independently of an actual problem schema, influence problem interpretation, and consequently, influence also solving strategies, learning and transfer between problems. The identification of these dimensions relies on studying isomorphic problems as well as on an accurate analysis of the strategies used by the pupils, whether they lead to a correct result or not. We believe that fundamental insight in understanding learning processes and modeling learners may be gained through studying a "relevant" micro domain in a detailed manner. Thus, even if our target is to enlarge in the long run the scope of exercises treated by DIANE, the range covered is not so crucial for us compared to the choice of the micro domain and the precision of the analysis. We consider as well that a data analysis at a procedural level is a prerequisite to more epistemic analyses: the automatic generation of a protocol analysis is a level of diagnostic that seems crucial to us and which is the one implemented in DIANE right now. It makes possible to test at a fine level hypotheses regarding problem solving and learning mechanisms with straightforward educational implications. Having introduced our theoretical background that stresses the importance of interpretive aspects and transverse semantic dimensions in arithmetical problem solving, we will then present the kind of problems we are working with, describe DIANE in more details and provide some results of experiments of cognitive psychology that we conducted.

1. Toward a semantic account of arithmetical problem solving

1.1 From schemas to mental models

The 80's were the golden age for the experimental works and the theories concerning arithmetical problem solving. The previously prevalent conception was that solving a story problem consisted mainly in identifying the accurate procedure and applying it to the accurate data from the problem. This conception evolved towards stressing the importance of the conceptual dimensions involved. Riley, Greeno, & Heller [10] established a typology of one-step additive problems, differentiating combination problems, comparison problems and transformation problems. Kinstch & Greeno [7] have developed a formal model for solving transformation problems relying on problem schemas. Later on, the emphasis on interpretive aspects in problem solving has led to the notion of the mental model of the problem introduced by Reusser [9], which is an intermediate step between reading the text of the problem and searching for a solution. This view made it possible to explain the role of some semantic aspects which were out of the scope of Kinstch & Greeno's [7] model; for instance, Hudson [6] showed that in a comparison problem, where a set of birds and a set of worms are presented together, the question *How many birds will not get a worm ?* is easier to answer than the more traditional form *How many more birds are there than worms ?,* and many studies have shown that a lot of mistakes are due to misinterpretations [4]. Thus, these researches emphasized the importance of two aspects: conceptual structure and interpretive aspects, which have to be described more precisely. Informative results come from works on analogical transfer.

1.2 Influence of semantic dimensions

More recently, work on analogical transfer showed that semantic features have a major role in problem solving process. Positive spontaneous transfer is usually observed when both semantic and structural features are common [1]. When the problems are similar in their surface features but dissimilar in their structure, the transfer is equally high but negative [11], [8]. Some studies have explicitly studied the role of semantic aspects and attributed the differences between some isomorphic problem solving strategies to the way the situations are encoded [2]. Several possibilities exist for coding the objects of the situation and a source of error is the use of an inappropriate coding, partially compatible with the relevant one [13].

Within the framework of arithmetic problems, our claim is that the variables involved in the problem are an essential factor that is transverse to problem schemas or problem types. We propose that the different types of quantities used in arithmetic problems do not behave in a similar way. Certain variables call for some specific operations. Quantities such as weights, prices, and numbers of elements may be easily added, because we are used to situations where these quantities are accumulated to give a unique quantity. In this kind of situations, the salient dimension of these variables is the cardinal one. Conversely, dates, ages, durations are not so easy to add: although a given value of age may be added to a duration to provide a new value of age; in this case, the quantities which are added are not of the same type. On the other hand, temporal or spatial quantities are more suited to comparison and call for the operation of subtraction, which measures the difference in a comparison. In this kind of situations, the salient dimension of these variables is the ordinal one.

We want to describe in a more precise way the semantic differences between isomorphic problems by characterizing their influence. For this purpose, it seems necessary to study problem solving mechanism at a detailed level which makes it possible to identify not only the performance but the solving process itself and to characterize the effect of the interpretive aspects induced by the semantic dimensions. Thus, we constructed a structure of problems from which we manipulated the semantic features.

2. A set of structured exercises and their solving models

Several constraints were applied in order to choose the exercises. (i) Concerning the conceptual structure, the part-whole dimension is a fundamental issue in additive problem solving; it appears as being a prerequisite in order for children to solve additive word problems efficiently [14]; thus our problems are focused on a part-whole structure. (ii) We looked for problems that could be described in an isomorphic manner through a change of some semantic dimensions. We decided to manipulate the variables involved. (iii) We looked for a variety of problems, more precisely problems that would allow the measure of the influence of the variable on the combination/comparison dimension. Hence, we built combination problems as well as comparison problems (iii) In order to focus on the role of transverse semantic dimensions, we looked for problems that did not involve either procedural or calculation difficulties. Therefore, we chose problems involving small numbers. (iv) We looked for problems allowing several ways to reach the solution so as to study not only the rate of success but the mechanisms involved in the choice of a strategy, whether it is adequate or not and to assess the quality of DIANE's diagnosis in non trivial situations. As a result, we built problems that might require several steps to solve.

The following problems illustrate how those constraints were embedded:
John bought a 8-Euro pen and an exercise book. He paid 14 Euros. Followed by one of these four wordings:
- *Paul bought an exercise book and 5-Euro scissors. How much did he pay?*
- *Paul bought an exercise book and scissors that costs 3 Euros less than the exercise book. How much did he pay?*
- *Paul bought an exercise book and scissors. He paid 10 Euros. How much are the scissors?*
- *Paul bought an exercise book and scissors. He paid 3 Euros less than John. How much are the scissors?*

Those problems have the following structure: all problems involve two wholes (Whole1 and Whole2) and three parts (Part1, Part2, Part3); Part2 is common to Whole1 and Whole2. The values of a part (Part1) and of a whole (Whole1) are given first (John bought a 8 Euros pen and an exercise book. He paid 14 Euros). Then, a new set is introduced, sharing the second part (Part2) with the first set. In the condition in which the final question concerns the second whole (Whole2) a piece of information is stated concerning the non common part (Part3), this information being either explicit (combination problems: Paul bought an exercise book and 5-Euro pair of scissors) either defined by comparison with Part1 (comparison problems: Paul bought *an exercise book* and scissors that cost 3 Euros less than the exercise book). In the condition in which the final question concerns the third part (Part3) a piece of information is stated concerning the second whole (Whole2), this information being either explicit (combination problems: Paul bought *an exercise book* and scissors. He paid 10 Euros) either defined by comparison with Whole1 (comparison problems: Paul bought an exercise book and scissors. He paid 3 Euros less than John). Then a question concerns the missing entity: Part 3 (How much are the scissors?) or Whole2 (How much did Paul pay?).

In fact, three factors were manipulated in a systematic manner for constructing the problems presented hereby:

- The nature of the variable involved.
- The kind of problem (2 modalities: complementation or comparison): if the problem can be solved by a double complementation, we call it a complementation problem; if it can be solved by a complementation followed by a comparison, we call it a comparison problem.
- The nature of the question (2 modalities: part or whole): If the question concerns Whole2, we call it a whole problem and if the question concerns Part3, we call it a part problem.

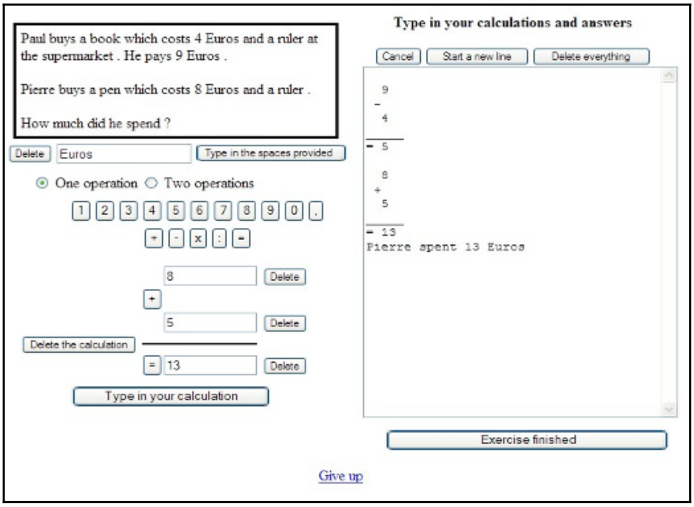
The two last factors define four families of problems that share some structural dimensions (two wholes, a common part and the explicit statement of Whole1 and Part1) but differ in others (the 2x2 previous modalities). Among each family, we built isomorphic problems through the use of several variables that we will describe more precisely later on.

One major interest of those problems is that they can all be solved by two alternative strategies that we named *step by step* strategy and *difference* strategy. The *step by step* strategy requires to calculate Part2 before determining whether Part3 or Whole2 (calculating that the price of the *exercise book* is 6 Euros in the previous example). The *difference* strategy does not require to calculate the common part and is based on the fact that if two sets share a common part, then their wholes differ by the same value as do the specific parts (the price of the pen and the price of the scissors differ by the same value as the total prices paid). It has to be noted that, if in complementation problems both strategies are in two steps, in the case of the comparison problem, the *step by step* strategy require three steps whereas the *difference* strategy requires only one. There exists as well a *mixed* strategy, that leads to the correct result even though it involves a non useful calculation; it starts with the calculation of Part 2 and ends with the *difference* strategy.

The solving model used for DIANE is composed of the following triple RM=(T, S, H). T refers to the problem Type and depends on the three parameters defined above (kind of problem, nature of the question, nature of the variable). S refers to the Strategy at hand (*step by step*, *difference* or *mixed* strategy). H refers to the Heuristics used and is mostly used to model the erroneous resolution; for instance applying an arithmetic operator to the last data of the problem and the result of the intermediate calculation.

3. Description of DIANE

DIANE is a web based application relying on open source technologies. DIANE is composed of an administrator interface dedicated to the researcher or the teacher and of a problem solving interface dedicated to the pupil. The administrator interface allows the user to add problems, according to the factors defined above, to create series of exercises, to look for the protocol of a student, or to download the results of a diagnosis. The role of the problem solving interface is to enable the pupil to solve a series of problems that will be analyzed later on and will be the basis for the diagnosis. This interface (Figure 1) provides some functions aimed at facilitating the calculation and writing parts of the process in order to let the pupil concentrate on the problem solving. The use of the keyboard is optional: all the problems can be solved by using the mouse only. The answers of the pupils are a mix of algebraic expressions and natural language. All the words which are necessary to write an answer are present in the text; the words were made clickable for this purpose. Using only the words of the problem for writing the solution helps to work with a restrained lexicon and avoids typing and spelling mistakes; it allows us to analyze a constrained natural language.



Paul buys a book which costs 4 Euros and a ruler at the supermarket . He pays 9 Euros .

Pierre buys a pen which costs 8 Euros and a ruler .

How much did he spend ?

Euros

● One operation ○ Two operations

8

+

5

= 13

Type in your calculations and answers

```
 9
-
 4
= 5
 8
+
 5
= 13
Pierre spent 13 Euros
```

Figure 1. *The pupil interface*

4. Diagnosis with DIANE

Diagnosis with DIANE is a tool for analyzing and understanding the behavior of the learners at a detailed level when they solve arithmetic problems. The diagnosis is generic in that it might be applied to all the classes of problems that are defined and is not influenced by the surface features of the exercises. Diagnosis concerns not only success or failure or the different kinds of successful strategies, but erroneous results are coded at the same detailed level as the successful strategies. As we have already mentioned, our main rationale is that understanding the influence of representation on problem solving requires the analysis of behavior at a very detailed level. Note that more than half of the modalities of the table of analysis are used for encoding errors.

Diagnosis is reported in a 18 column table. Depending on the strategies and the nature of the problem up to 14 columns are effectively used for analyzing one particular resolution. The first column encodes the strategy. It is followed by several groups of four columns. The first column of each group encodes the procedure (addition, subtraction, etc), the second one indicates whether the data are relevant, the third one indicates whether the result is correct and the fourth one indicates whether a sentence is formulated and evaluates the sentence (this column is not yet encoded automatically). Another column, the 14th is used to identify the nature of what is calculated in the last step of the resolution (a part, a whole, the result of a comparison, an operation involving the intermediary result and the last item of data, etc.)

The answer of the pupil, a string of characters, is treated following the pattern of regular expressions. This treatment turns the answer of the pupil into four tables, which are used for the analysis. The first table contains all the numbers included in the answer, the second one contains all the operations, the third one all numbers that are not operands and the fourth one contains all the words separated by spaces.

The data extracted or inferred from the problem (Whole1, Part1, Part3 …) are stored in a database. The automatic diagnosis is based on comparisons between the data extracted and inferred from the text and the tables, through using heuristics derived from the table of analysis.

The following table (Table 1) provides two examples of diagnosis for the problem: *John bought a 8-Euro pen and an exercise book. He paid 14 Euros. Paul bought an exercise book and scissors. He paid 3 Euros less than John. How much are the scissors?*

Pupil 1		Pupil 2	
Response	Diagnosis by DIANE	Response	Diagnosis by DIANE
14 - 8 = 7 14 - 3 = 11 11 - 7 = 4 The scissors cost 4 Euros	Col 1: *step by step* strategy Col 2-4: subtraction, relevant data, calculation error Col 6-8: subtraction, relevant data, exact result Col 14: calculation of a part Col 15-17: subtraction, relevant data (the calculation error is taken into account), exact result	14 - 8 = 6 14 - 3 = 11 The scissors cost 11 Euros	Col 1: Erroneous *comparison* strategy Col 2-4: subtraction, relevant data, exact result Col 14: calculation of comparison Col 15-17: subtraction, data correct for the comparison but not for the solution, exact result

Table 1: An example of Diagnosis with DIANE

DIANE provides a fine grained diagnosis that identifies the errors made by the pupils. For instance, pupil 1 (Table 1) made a calculation mistake when calculating Part 2 (14-8=7), which implies an erroneous value for the solution (11-7=4). DIANE indicates that an item of data is incorrect in the last calculation due to a calculation error at the first step. The same holds true for erroneous strategies. Pupil 2 (Table 1), after having performed a correct first step ends his/her resolution with the calculation of the comparison (14-3=11). In this situation, DIANE diagnosis indicates that the pupil used an erroneous strategy that provided a result which is correct for the calculation of the comparison but not for the solution. This situation is a case of use of the heuristic previously described (using the last data and the result of the intermediate calculation).

5. Results from experimental psychology

Experimentation has been conducted on a large scale [12]; 402 pupils (168 5th graders, 234 6th graders) from 15 schools in Paris and the Toulouse area participating. The experimental design was the following: each child solved, within two sessions, complementation and comparison problems for three kinds of variables and the two kinds of questions, that is twelve problems. Even if the experimental results are not the main scope of this paper, let us mention that the main hypotheses were confirmed (for each of the four families of problems, we found a main effect of the kind of variable on the score of success ($17,79<F(2, 401)<51,12$; $p<0.0001$ for all the analyses). As predicted, we also found that cardinal variables made combination problems easier and ordinal variables made comparison problems easier. Furthermore, similar results were observed concerning the strategies at hand: strategies were highly dependent on the variable involves. For instance, in a comparison problem in which the variable was an age, 64% of the pupils used a strategy that did not require to calculate the intermediate part. Conversely, for the isomorphic problem in which the variable was a price, only 4% did so. We were also able to generalize our results to a larger scale of variables [5]. The table of analysis, on which DIANE's diagnosis is based was tested manually on those protocols. Except that human coding requiring a long training period for the coder, was slow and difficult, results were very satisfactory: (i) between judge agreement was always more than 95% for well trained coders for all the samples that we tested, and (ii) the detailed level of description made it possible to distinguish between and to embrace a large variety of behaviors.

6. Assessment of the quality of DIANE's diagnosis

In order to assess the quality of the automatic diagnosis, we carried out two experiments.

For the first one, we typed the protocols issued from a pen and pencil experiment in a 5th grade class [12] with 29 pupils. Each protocol included 12 problems, thus we analyzed 308 productions. In the second one, the experimentation was conducted directly with the interface and concerned 46 pupils from one 5th grade class and one 6th grade class. Each of the children solved 6 problems in this situation [3] and we analyzed 276 productions. For this second situation we might note that no difficulty due to the use of the interface was identified neither by the children nor by the experimenter; the interface was very easily used and well accepted by the children. The main experimental measures provided no significant results concerning the success rate or the strategy used between the two experiments [3]. However, the question of the difference of behavior between the pen and pencil situation and the interface situation will be looked at more deeply in forthcoming studies.

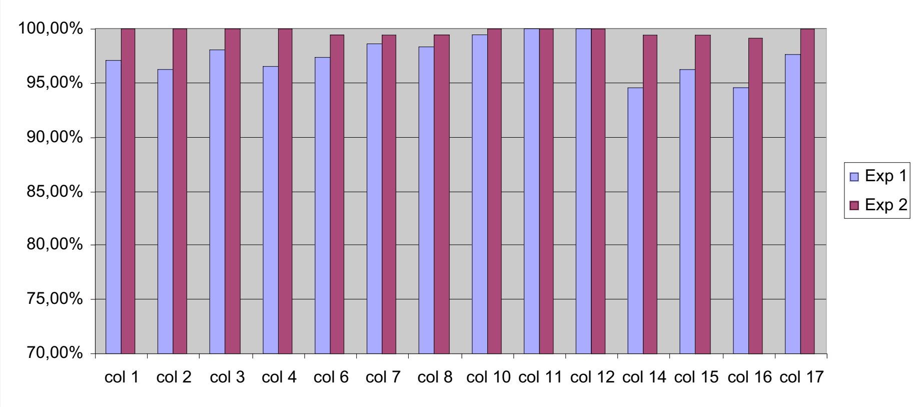


Figure 2. Distribution of rates of equality between automatic and manual coding

Figure 2 illustrates that for all the columns, and for each of the two experiments, the rate of equality between the manual encoding and the automatic one was between 94.5 and 100%. Furthermore, for 93.4% of the problems, automatic encoding is equal to manual encoding for all the columns encoded. Thus, these two experiments confirmed that DIANE is actually able to make a diagnosis of a quality close to the manual one.

7. Conclusion and perspectives

In this paper, we introduced DIANE, an environment aimed at diagnosing at a detailed level arithmetical word problems and currently specialized in four families of two-steps additive combination and comparison problems. These problems were designed in order to make it possible to test hypotheses on cognitive mechanisms involved in arithmetical problem solving that have direct educational implications. We are now working in three directions with DIANE. (i) We want to build a fine-grained typology of the strategies based on DIANE's diagnosis that will serve as a basis for the remediation module. We already constructed [12] a manual typology that includes all the successful strategies, and nearly 80% of the erroneous ones. (ii) We want to produce diagnoses that are straightforwardly understandable by teachers. The diagnosis produced by DIANE provides information on the solving process that teachers found very informative. We are now working on a module that will produce the diagnosis in natural language. (iii) We plan to enlarge the range of problems considered by DIANE: all the one-step additive word problems can be nearly readily integrated, and this diagnosis will be involved in the remediation module.

References

[1] Susan M. BARNETT; Stephen J. CECI (2002). When and where do we apply what we learn? A taxonomy for far transfer. Psychological bulletin, vol. 128, no 4, pp. 612 - 637

[2] Bassok, M. Wu, L. & Olseth, L.K. (1995), Judging a book by its cover: Interpretative effects of content on problem solving transfer. Memory & Cognition, 23, 354-367.

[3] Calestroupat, J., Catégorisation d'interprétations conduisant à des erreurs dans la résolution de problèmes arithmétiques, Mémoire de DEA, Université de Paris 8, 2004.

[4] Cummins, D.D., Kintsch, W., Reusser, K. & Weimer, R. «The role of understanding in solving words problems». *Cognitive Psychology*, vol. 20, p. 405-438.

[5] Gamo, S. Aspects sémantiques et rôle de l'amorçage dans la résolution de Problèmes additifs à étapes. Mémoire de DEA. Université de Paris 8

[6] Hudson, T. (1983). Correspondences and numerical differences between disjoint sets. Child Development, 54, 84-90.

[7] Kintsch, W. & Greeno, J.G. (1985). Understanding and solving word arithmetic problems. Psychological Review, 92,1, 109-129.

[8] Novick, L. (1988), Analogical transfer, problem similariry, and expertise. Journal of Experimental Psychology: Learning, Memory, & Cognition, 14, 510-520.

[9] Reusser, K., (1989). From text to situation to equation Cognitive simulation of understanding and solving mathematical word problems. European Journal in an International Context, 2, 2, 477-498.

[10] Riley, M.S., Greeno, J.G. &Heller, J.I. (1983). Development of problem solving ability in arithmetic. In H.P. Ginsberg(Ed.)

[11] Ross, B.H. (1989), Distinguishing types of superficial similarities: Different effects on the access and use of earlier problems. Journal of Experimental Psychology: Learning, Memory, & Cognition, 15, 456-468.

[12] Sander, E., Levrat, B., Brissiaud, R., Porcheron, P., Richard, R (2003). Conceptualisation et propriétés sémantiques des situations dans la résolution de problèmes arithmétiques : rapport d'étape. Ministère de la Recherche ; appel d'offre 2002 Ecole et Sciences Cognitives : les apprentissages et leurs dysfonctionnements.

[13] Sander, E., & Richard, J.-F. (1998). Analogy making as a categorization and an abstraction process. In K. Holyoak, D. Gentner, & B. Kokinov (Eds.) Advances in analogy research: Integration of theory and data from the cognitive, computational and neural sciences (pp. 381-389). Sofia: NBU Series in Cognitive Sciences.

[14] Sophian, C, & McCorgray, P. (1994) Part-whole knowledge and early arithmetic problem solving. Cognition and Instruction, 12, 1, 3-33

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Collaboration and Cognitive Tutoring: Integration, Empirical Results, and Future Directions

Andreas HARRER[+], Bruce M. MCLAREN[*],
Erin WALKER[*], Lars BOLLEN[+], Jonathan SEWALL[*]

[+]*University Duisburg-Essen, Duisburg, Germany*
{harrer, bollen}@collide.info
[*]*Carnegie Mellon University, Pittsburgh, PA USA*
{bmclaren, sewall}@cs.cmu.edu, erinwalk@andrew.cmu.edu

Abstract. In this paper, we describe progress we have made toward providing cognitive tutoring to students within a collaborative software environment. First, we have integrated a collaborative software tool, Cool Modes, with software designed to develop Cognitive Tutors (the Cognitive Tutor Authoring Tool). Our initial integration provides a means to capture data that acts as the foundation of a tutor for collaboration but does not yet fully support actual tutoring. Second, we've performed two exploratory studies in which dyads of students used our software to collaborate in solving modelling tasks. These studies uncovered five dimensions of observed behavior that point to the need for abstraction of student actions to better recognize, analyze, and correct collaborative steps in problem solving. We discuss plans to incorporate such analyses into our approach and to extend our tools to eventually provide tutoring of collaboration.

1. Introduction

Cognitive Tutors, a particular type of intelligent tutor that supports "guided learning by doing" [1], have been shown to improve learning in domains like algebra and geometry by approximately one standard deviation over traditional classroom instruction [2]. So far, cognitive tutors have been used only for one-on-one instruction—a computer tutor assisting a single student. We seek to determine whether a cognitive tutoring approach can support and improve learning in a collaborative environment.

Collaboration is recognized as an important forum for learning [3], and research has demonstrated its potential for improving students' problem-solving and learning [e.g., 4, 5]. However, collaboration is a complex process, not as constrained as individual learning. It raises many questions with respect to cognitive tutoring: Can a single-student cognitive model be extended to address collaboration? Can a cognitive tutor capture and leverage the data available in a collaborative scenario, such as chat between mutiple students? What types of collaborative problems are amenable to a cognitive tutoring approach?

To take a step toward addressing these questions, we have integrated and begun experimentation with a collaborative work environment and a cognitive tutoring tool [6]. Our initial goals are twofold. First, we capture and analyze data from live collaboration so that we can better understand how a cognitive tutor might use that data to diagnose and tutor student action in a collaborative environment. Second, we would eventually like to directly use the data we collect as the basis for the cognitive tutor model.

To that end, we have developed an approach called bootstrapping novice data (BND) in which groups of students attempt to solve problems with a computer-based collaborative tool. While they work, the system records their actions in a network representation that combines all collaborating groups' solutions into a single graph that can be used for analysis and as the basis for a tutor. To effect the BND approach we have combined two software tools: a collaborative modeling tool, Cool Modes (Collaborative Open Learning and MODEling System) [7], and a tutor authoring environment, the Cognitive Tutor Authoring Tools (CTAT) [8]. Our work has focused on data collection and analysis; actual tutoring in the collaborative context is yet to be done but will be guided by these initial findings.

In this paper, we illustrate how we have implemented the BND methodology, describe empirical work that explores a particular type of collaborative problem and tests the BND approach, and present our ideas for extending our approach both to improve analysis and to lead to our ultimate goal of providing tutoring in a collaborative environment.

2. Realization of BND: The Integration of Cool Modes and the Behavior Recorder

In our implementation, depicted in Figure 1, Cool Modes (shown on the left) provides the user interface for the student; it includes a shared workspace that all collaborating students in a session can view and update, a palette with objects that users can drag onto the workspace, a chat area, and a private workspace. Cool Modes sends messages describing students' actions (e.g., "student *A* created classification link *L*") to CTAT's Behavior Recorder (or "BR," shown on the right of Figure 1), which stores the actions in a *behavior graph.* Each edge in the graph represents a single student action, and paths through the graph represent series of student actions.

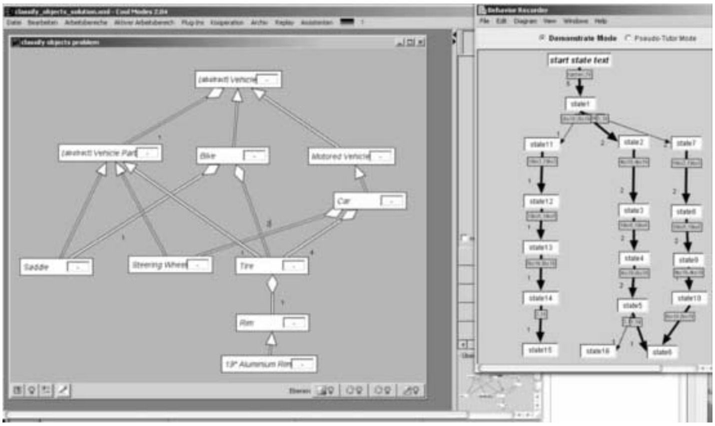


Figure 1: The student's view of the integrated Cool Modes (left) and the Behavior Recorder (right) environment. This shared Cool Modes workspace is from a vehicle classification / composition task. The behavior graph at right shows the amalgamated solutions of different collaborating groups of students.

A key aspect of the BND approach is that it counts the number of times actions are taken and displays these counts on the edges of the behavior graph. Thus, after a number of groups have used the integrated system, the behavior graph contains the actions of all student groups and reveals the frequency of common paths, both correct and incorrect. Use of this actual novice data can help to avoid part of the "expert blind spot" problem, in which experienced problem-solvers and teachers fail to identify common errors of novice students [9]. A tutor author can then use the BR to create a problem-specific tutor (or pseudo tutor, [8]) directly from the graph by labeling edges with hints and buggy messages.

We have integrated Cool Modes and the BR in a loosely-coupled fashion. Both tools remain fully operational by themselves, but can exchange messages bidirectionally using the MatchMaker communication server [10] and a "Tutor Adapter" (see Figure 2). Our earlier implementation provided one-way communication, which could support the recording of student actions but not tutoring [6]. Now, a student action causes the Cool Modes client to send an event to the MatchMaker server, which sends this event to the Tutor Adapter, which in turn forwards the event to the BR. If an author were to create a pseudo tutor and switch the BR from recording to tutoring mode, then it would respond to incoming events by sending bug messages and hints to the appropriate student or students.

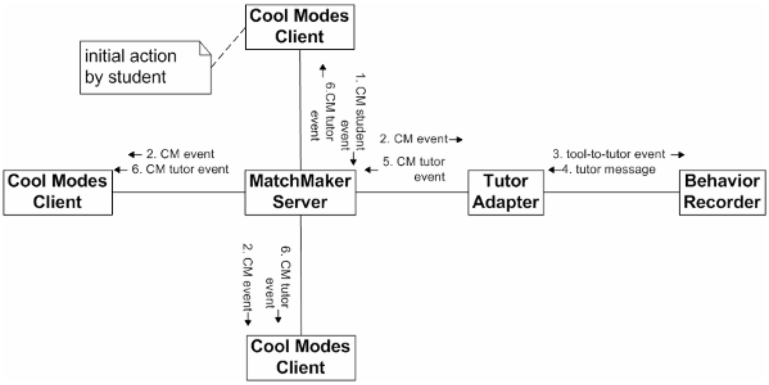


Figure 2: Collaboration diagram showing the message flow between Cool Modes and Behavior Recorder.

There are two key advantages to the BND approach. First, direct capture of student data for use in tutor building is a powerful idea. While student data has been used to guide tutor design [11] and tune tutor parameters [12], it has not been used directly as input for building an intelligent tutor. The potential time savings in data collection, data analysis, and tutoring with a single integrated tool could be significant. Second, given the complexity of collaborative learning, we thought that a 2-D visualization, in the form of a behavior graph, might allow for a better understanding and analysis of collaborative behavior when compared with, for instance, a non-visual, linear representation such as production rules.

3. Using the Behavior Recorder to Analyze Collaboration

The BR was originally designed for single-student tutoring of well-defined problems (e.g., mathematics, economics), which tend to have less possible correct and incorrect actions. In more open-ended collaborative problems, however, there are many possible sequences and alternative actions, and a given action may be appropriate in one context but not another. In this situation, a single behavior graph containing student actions is hard to interpret because higher-level processes like setting subgoals are not represented, and it is difficult to compare solutions, since on an action-by-action level most solutions will appear to be completely different. Additionally, larger group sizes also increase the state space of the Behavior Graph, because of different, yet potentially semantically equal sequences of actions by different users. Thus, early on it appeared to us that the BR would need to be extended using multiple levels of abstraction to handle the increased complexity of collaborative actions.

In preliminary experimentation with Cool Modes collaboration, we were able to identify five common dimensions of student action: *conceptual understanding, visual organization, task coordination, task coherence,* and *task selection.* Conceptual understanding refers to a pair's ability to successfully complete the task, while visual organization refers to a pair's ability to visually arrange the objects involved in an appropriate manner. Task coordination refers to skills in coordinating actions in the problem, without reference to the content of the actions. It includes sharing the work between all group members, and knowing what type of action to take at a given time (i.e., knowing when it is a good idea to reorganize the objects involved in the problem). Task coherence refers to the strategic appropriateness of the content of student actions, dealing with both task-oriented content (i.e., do adjacent phases of action deal with the appropriate objects) and collaborative content (i.e., are students providing good explanations to each other). Finally, task selection refers to students' abilities to set task-oriented and collaborative subgoals for solving the problem.

In order for the BR to process these five dimensions, it needs to handle actions at different levels of abstraction. Conceptual understanding and visual organization can be dealt with on an action-by-action basis. On the other hand, task coordination and task coherence are best evaluated through the analysis of *phases* of action, or chains of the same type of action. A chain of chat actions followed by chain of creation actions would indicate that, on a task coordination level, students have decided to discuss what objects they should create and then create some objects. This type of information is difficult, if not impossible, to extract from an action-by-action representation. Finally, task selection can be analyzed in the BR by aggregating multiple phases of action which represent high-level goals.

4. Empirical Studies

We performed two experiments to explore our assessment of the information required by the BR. Each experiment involved a visual modelling problem and tested the effect of the initial organization of objects on the collaborative problem-solving effort. In Experiment 1, we established these five elements of collaboration as relevant to the Cool Modes classification problem, and showed the need for adding support for different levels of abstraction to the BR. In Experiment 2, we verified that the five elements of collaboration are generalizable to a CoolModes Petri Net problem, and explored how the five elements could be analyzed and tutored using the BR. We will summarize the results of Experiment 1 (for a more detailed description see [13]) and describe the results of Experiment 2 in detail.

4.1 Experiment 1

In this experiment we asked 8 dyads of students to collaborate on solving a classification / composition problem (depicted in Figure 1). Students could take three types of actions: *chat* actions, "talking" to a partner in a chat window, *move* actions, repositioning an object in the shared workspace, and *creation/deletion* actions, creating or deleting links between objects. There were two conditions: in the *ordered* condition, the initial presentation showed related objects visually close to one another, to provide a well-organized display of the desired final network; in the *scrambled* condition, objects were positioned randomly. Groups 1 to 5 were in the scrambled condition; groups 6 to 8 were in the ordered condition. The results of the first experiment are summarized in Table 1.

The five dimensions of analysis illustrated positive and negative strategies of the participants as they related to the quality of the final solutions. Additionallly, the dimensions highlighted the connection between the organization of the start state and participants' conceptual understanding and collaborative processes.

Table 1: Solution Types and Dimensions of Analysis

	Groups 5 and 8	*Groups 2,6, and 7*	*Groups 1, 3, and 4*
Conceptual Understanding	**Good** – only trivial mistakes	**Incomplete** – only one link extended from each class	**Inconsistent** – too many links extended from each class
Visual Organization	**Good** - based on abstractions	**Overly organized** – had a tree-like structure	**Disorganized** – had long, intersecting links
Task Coordination	**Good** – good alternation of phases and distribution of work	**Hesitant –** long chat phases, formal turn-taking structure	**Impulsive –** creation before organization, informal turn-taking.
Task Coherence	**Good** - adjacent phases referred to similar objects and levels of abstraction.	**Good** - adjacent phases referred to similar objects and levels of abstraction.	**Poor** – adjacent phases referred to different objects
Task Selection	**Good** - based on abstractions	**Good** - based on abstractions	**Poor** - based on visual proximity

4.2 Experiment 2

We asked 8 dyads to solve a traffic light modelling problem using the Cool Modes / BR integrated system. Students were asked to model the coordination of car and pedestrian lights at a given intersection using Petri Nets (i.e., they were asked to draw links between traffic lights and transitions). Students could take chat, move, and creation/deletion actions, as in Experiment 1, but also *simulation* actions, firing transitions to move from one state to another. In the ordered condition of Experiment 2, the objects were organized like real-world traffic lights, with the car lights on one side, the pedestrian lights on the other side, and the transitions in the middle. In the scrambled condition, objects were placed randomly in the workspace.

We were again able to analyze the results using the five dimensions. To evaluate *conceptual understanding*, solutions were rated on a 9-point scale based on the requirements of the problem (e.g., during a simulation, the solution should never have pedestrians and cars moving at the same time). The scrambled group had significantly better solutions than the ordered group (Ms = 5.25 and 1.75). Solutions could be further divided into good (groups 1 and 2, M = 6.5), mediocre (groups 3, 4, and 5, M = 3.7), and poor (groups 6, 7, and 8, M = 1.3). The scrambled group had two good and two medium solutions, and the ordered group had one medium and three bad solutions.

The *visual organization* of the final solutions can be described in terms of two competing schemes: "real-world" (i.e., separating the car and pedestrian lights and arranging them in red/yellow/green order) versus “easy-to-follow” (i.e., having minimal edge crossings). A real-world scheme meant that the best place for the transition links were in the center of the shared visual space, creating confusing solutions because links intersected and extended in many different directions. In the ordered start state, the ideal solution corresponded to the real world, but was not easy-to-follow. Three out of the four ordered groups did not significantly reposition the objects from their original places in the start state. On the other hand, all four of the groups in the scrambled condition moved objects from their initial disorganized state to good final solutions that were relatively easy to follow. It appears that our conception of an "organized" condition may not have been as well founded for this particular problem, since an easy-to-follow arrangement seemed to relate to better solutions than a real-world arrangement.

The results for the *task coordination* differed significantly between good and bad solutions. Good groups had a significantly fewer percentage of chat actions than mediocre and poor groups (Ms = 12%, 48%, and 44%), and a significantly lower percentage of chat phases (Ms = 20%, 40%, and 39%). The good groups and the two mediocre groups in the scrambled condition also had a significantly higher percentage of move actions than the ordered groups (Ms = 28% and 8%) and significantly more move phases (Ms = 23% and 11%). There was some statistical evidence that the ordering of phases also had an effect on whether groups did well or poorly, with the optimal sequence of phases being chat->move->creation/deletion->simulation. Further, the good groups had a less balanced work distribution than the mediocre and poor groups. The ordered (and therefore less successful) groups split their time between having one person perform the whole phase (M = 37%), the other person perform the whole phase (M = 34%), or both people taking action in the phase (M = 28%). The scrambled groups had fewer phases where both people took action (M = 15%), and a less balanced distribution of individual phases (Ms = 53% and 32%). These results were surprisingly congruent with the task coordination results for Experiment 1, as reported in detail in [13].

Although *task coherence* varied between conditions in Experiment 1, there were few differences on this dimension between groups in Experiment 2. Groups refered to an average of 1.8 objects per phase in move phases, creation/deletion phases, and simulation phases. All groups tended to refer to the same objects across multiple phases.

Task selection also did not differ between groups in this experiment, but commonalities between groups provided insight into the collaborative process. Groups structured their actions based on the transitions from one state of traffic lights to the next. Creation/deletion actions were linear 79% of the time, in that the current edge being drawn involved an object used in the previous creation/deletion action. Groups tended to focus on either the pedestrian or the car lights at a given time; the current creation/deletion action tended to involve the same light class as the previous creation/deletion action 75% of the time.

In addition to the analysis of Experiment 2 based on the five dimensions, we explored how the BR could be used to analyze and tutor collaboration. For example, we used the BR to capture individual creation actions, and discovered that two groups (1 and 3) used the same correct strategy in creating the links necessary to have the traffic lights turn from green to yellow to red. This path in the graph demonstrated a conceptual understanding of how Petri Nets can be used to effect transitions. We will ultimately be able to add hints that encourage students to take this path, leveraging the behavior graph as a means for tutoring. In likewise fashion, the BR can also be used to identify common bugs in participants' action-by-action problem solving. For instance, the BR captured a common error in groups 1 and 2 of Experiment 2: each group built a Petri Net, in almost identical fashion, in which the traffic-red and pedestrian-green lights would not occur together. In situations like this, the behavior graph could be annotated to mark this sequence as buggy, thus allowing the tutor to provide feedback should a future student take the same steps.

On the other hand, it is clear that the level of individual actions is not sufficient for representing all of the dimensions. For instance, evaluating whether students are chatting "too much" or alternating phases in an "optimal" way is not easily detected at the lowest level of abstraction. To explore how we might do more abstract analysis, we wrote code to pre-process and cluster the Cool Modes logs at a higher level of abstraction and sent them to the BR. Figure 3 shows an example of this level of analysis from Experiment 2. Instead of individual actions, edges in the graph represent phases of actions (see the "CHAT", "MOVE", and "OBJEC" designations on the edges). The number to the right of each phase in the figure specifies how many instances of that particular action type occurred during consecutive steps, e.g., the first CHAT phase, starting to the left from the root node, represents 2 individual chat actions. The graph shows the first 5 phases of groups 2, 3, 5, and 8. Because the type of phase, the number of actions within each phase, and who participates (recorded but not shown in the figure), is recorded we can analyze the data and, ultimately, may be able to provide tutor feedback at this level. For instance, notice that the scrambled groups (2 and 3) incorporated move phases into their process, while at the same point, the organized groups (5 and 8) only used CHAT and OBJEC (i.e., creation/deletion) phases. Additionally, groups 5 and 8 began their collaboration with a lengthy chat phase, and group 5 continued to chat excessively (23 chat actions by group 5 leading to state22!). This level of data provided to the BR could help us to understand better the task coordination dimension. In addition, if provided at student time, the BR could also provide feedback to groups with "buggy" behavior; for instance, a tutor might have been able to intervene during group 5's long chat phase. In future work, we intend to further explore how this and other levels of abstraction can help us address not only the task coordination dimension but also the task coherence and task selection dimensions.

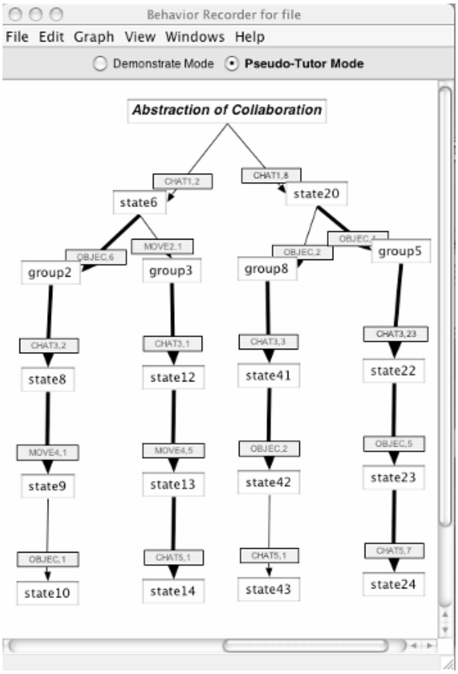


Figure 3. An Abstracted Behavior Graph

4.3 Discussion

There are two questions to answer with respect to these empirical results: Were the five dimensions valid units of analysis across the experiments? Can the BR analyze the dimensions and, if not, can the dimensions be used to guide extensions to it? The dimensions did indeed provide a useful analysis framework. The conceptual understanding dimension was helpful in evaluating problem solutions; in both experiments we were able to identify and rate the dyads based on salient (but different) conceptual features. Visual organization was important in both tasks, and appeared to inform problem solutions. The task coordination dimension provided valuable data, and the clearest tutoring guidelines of all the dimensions. The task coherence dimension provided information about object references in Experiment 1, but was not as clear of an aid in the analysis of Experiment 2. Finally, the task selection dimension was a useful measure in both experiments, but was more valuable in Experiment 1 due to the greater number of possible strategies.

With the introduction of abstraction levels, the effort to provide hints and messages to links will be greatly reduced because of the aggregation of actions to phases and sequences of phases. Even with abstraction, larger collaboration groups would naturally lead to greater difficulty in providing hints and messages, but our intention is to focus on small groups, such as the dyads of the experiments described in this paper.

5. Conclusion

Tackling the problem of tutoring a collaborative process is non-trivial. Others have taken steps in this direction (e.g., [14, 15]), but there are still challenges ahead. We have been working on capturing and analyzing collaborative activity in the Behavior Recorder, a tool for building Pseudo Tutors, a special type of cognitive tutor that is based on the idea of recording problem solving behavior by demonstration and then tutoring students using the captured model as a basis. The work and empirical results we have presented in this paper has led us to the conclusion that BR analysis needs to take place at multiple levels of abstraction to support tutoring of collaboration.

Using the five dimensions of analysis as a framework, we intend to continue to explore ways to analyze and ultimately tutor collaborative behavior. We briefly demonstrated one approach we are exploring: clustering of actions to analyze phases (of actions) and sequences of phases. Since task coordination appears to be an interesting and fruitful analysis dimension, we will initially focus on that level of abstraction. Previously, in other work, we investigated the problem of automatically identifying phases by aggregating similar types of actions [16] and hope to leverage those efforts in our present work. An architectural issue will be determining when to analyze (and tutor) at these various levels of abstraction. Another direction we have considered is extending the BR so that it can do "fuzzy" classifications of actions (e.g., dynamically adjusting parameters to allow behavior graph paths to converge more frequently).

We are in the early stages of our work but are encouraged by the preliminary results. We plan both to perform more studies to verify the generality of our framework and to implement and experiment with extensions to the Behavior Recorder.

References

[1] Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *Journal of the Learning Sciences*, *4*, 167-207.

[2] Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education, 8*, 30-43.

[3] Bransford, J. D., Brown, A. L., , & Cocking, R. R. (Eds.). (2000). *How people learn: Brain, mind, experience, and school.* Washington, DC: National Academy Press.

[4] Slavin, R. E. (1992). When and why does cooperative learning increase achievement? Theoretical and empirical perspectives. In R. Hertz-Lazarowitz & N. Miller (Eds.), *Interaction in cooperative groups: The theoretical anatomy of group learning* (pp. 145-173). New York: Cambridge University Press.

[5] Johnson, D. W. and Johnson, R. T. (1990). Cooperative learning and achievement. In S. Sharan (Ed.), *Cooperative learning: Theory and research* (pp. 23-37). New York: Praeger.

[6] McLaren, B. M., Koedinger, K. R., Schneider, M., Harrer, A., & Bollen, L. (2004b) Toward Cognitive Tutoring in a Collaborative, Web-Based Environment; *Proceedings of the Workshop of AHCW 04*, Munich, Germany, July 2004.

[7] Pinkwart, N. (2003) A Plug-In Architecture for Graph Based Collaborative Modeling Systems. In U. Hoppe, F. Verdejo & J. Kay (eds.): *Proceedings of the 11th Conference on Artificial Intelligence in Education*, 535-536.

[8] Koedinger, K. R., Aleven, V., Heffernan, N., McLaren, B. M., & Hockenberry, M. (2004) Opening the Door to Non-Programmers: Authoring Intelligent Tutor Behavior by Demonstration. In *Proceedingsof ITS*, Maceio, Brazil, 2004.

[9] Nathan, M., Koedinger, K., and Alibali, M. (2001). Expert blind spot: When content knowledge eclipses pedagogical content knowledge. Paper presented at the *Annual Meeting of the AERA*, Seattle.

[10] Jansen, M. (2003) Matchmaker - a framework to support collaborative java applications. In the *Proceedings of Artificial Intelligence in Education* (AIED-03), IOS Press, Amsterdam.

[11] Koedinger, K. R. & Terao, A. (2002). A cognitive task analysis of using pictures to support pre-algebraic reasoning. In C. D. Schunn & W. Gray (Eds.), *Proceedings of the 24th Annual Conference of the Cognitive Science Society*, 542-547.

[12] Corbett, A., McLaughlin, M., and Scarpinatto, K.C. (2000). Modeling Student Knowledge: Cognitive Tutors in High School and College. *User Modeling and User-Adapted Interaction*, 10, 81-108.

[13] McLaren, B. M., Walker, E., Sewall, J., Harrer, A., and Bollen, L. (2005) Cognitive Tutoring of Collaboration: Developmental and Empirical Steps Toward Realization; Proceedings of the *Conference on Computer Supported Collaborative Learning*, Taipei, Taiwan, May/June 2005.

[14] Goodman, B., Hitzeman, J., Linton, F., and Ross, H. (2003). Towards Intelligent Agents for Collaborative Learning: Recognizing the Role of Dialogue Participants. In the *Proceedings of Artificial Intelligence in Education* (AIED-03), IOS Press, Amsterdam.

[15] Suthers, D. D. (2003). Representational Guidance for Collaborative Learning. In the *Proceedings of Artificial Intelligence in Education* (AIED-03), IOS Press, Amsterdam.

[16] Harrer, A. & Bollen, L. (2004) Klassifizierung und Analyse von Aktionen in Modellierungswerkzeugen zur Lernerunterstützung. In Workshop-Proc. Modellierung 2004 . Marburg, 2004.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Personal Readers: Personalized Learning Object Readers for the Semantic Web [1]

Nicola Henze [a,2]
[a] *ISI – Semantic Web Group,*
University of Hannover & Research Center L3S

Abstract. This paper describes our idea for personalized e-Learning in the Semantic Web which is based on configurable, re-usable personalization services. To realize our ideas, we have developed a framework for designing, implementing and maintaining personal learning object readers, which enable the learners to study learning objects in an embedding, personalized context. We describe the architecture of our *Personal Reader framework*, and discuss the implementation of personalization services in the Semantic Web. We have realized two Personal Readers for e-Learning: one for learning Java programming, and another for learning about the Semantic Web.

Keywords. web-based learning platforms & architectures adaptive web-based environments, metadata, personalization, semantic web, authoring

1. Introduction

The amount of available electronic information increases from day to day. The usefulness of information for a person depends on various factors, among them are the timely presentation of information, the preciseness of presented information, the information content, and the prospective context of use. Clearly, we can not provide a measurement for the expected utility of a piece of information for *all* persons which access it, nor can we give such an estimation for a single person: the expected utility varies over time: what might be relevant at some point might be useless in the near future, e.g. the information about train departure times becomes completely irrelevant for planning a trip if the departure time lies in the past. With the idea of a Semantic Web [2] in which machines can understand, process and reason about resources to provide better and more comfortable support for humans in interacting with the World Wide Web, the question of personalizing the interaction with web content is at hand: Estimating the individual requirements of the user for accessing the information, learning about a user's needs from previous interactions, recognizing the actual access context, in order to support the user to retrieve and access the part of information from the World Wide Web which fits best to his or her current, individual needs.

[1] This work has partially been supported by the European Network of Excellence REWERSE - Reasoning on the Web with Rules and Semantics (www.rewerse.net).

[2] Correspondence to: Nicola Henze, ISI - Semantic Web Group, University of Hannover & Research Center L3S, Appelstr.4, D-30167 Hannover Tel.: +49 511 762 19716; Fax: +49 511 762 19712; E-mail: henze@l3s.de

The development of a Semantic Web has, as we believe, also great impact on the future of e-Learning. In the past few years, achievements in creating standards for learning objects (for example the initiatives from LOM (Learning Objects Metadata [13]) or IMS [12]) have been carried out, and large learning object repositories like Ariadne [1], Edutella [7] and others have been built. This shifts the focus from the more or less closed e-Learning environments forward to open e-Learning environments, in which learning objects from multiple sources (e.g. from different courses, multiple learning object providers, etc.) could be integrated into the learning process. This is particularly interesting for university education and life-long learning where experienced learners can profit from self-directed learning, exploratory learning, and similar learning scenarios.

This paper describes our approach to realize personalized e-Learning in the Semantic Web. The following section discusses the theoretical background of our approach and motivates the development of our Personal Reader framework. The architecture of the Personal Reader framework is described in Section 3; here we also discuss authoring of such Personal Learning Object Readers as well as required annotations of of learning objects with standard metadata for these Readers. Section 4 shows the implementation of some example personalization services for e-Learning. Section 4.4 finally provides information about realized Personal Learning Object Readers for Java programming and Semantic Web.

2. Towards personalized e-Learning in the Semantic Web

Our approach towards personalized e-Learning in the Semantic Web is guided by the question how we can adapt personalization algorithms (especially from field of *adaptive educational hypermedia*) in a way that they can be

1. re-used, and
2. can be plugged together by the learners as they like - thus enabling learners to choose which kind of personalized guidance and in what combination they appreciate personalized e-Learning.

In a theoretical analysis and comparison of existing adaptive educational hypermedia systems that we have done in earlier work [10], we found that it is indeed possible to describe personalization functionality in a manner required for re-use, i.e. describe such personalization functionality in encapsulated, independent modules. Brusilovsky has argued in [5], that current adaptive educational hypermedia systems suffer from the so-called *open corpus problem*. Hereby is meant, that these systems work on a fixed set of documents/resources which are normally known to the system developers at design time. Alterations in the set of documents like modifying a document's content, adding new documents, etc., are nearly impossible because they require substantial alterations on the document descriptions, and normally affect relations in the complete corpus. To analyze the open-corpus-problem in more detail, we started in [10] an analysis of existing adaptive educational hypermedia systems and proposed a logic-based definition of adaptive educational hypermedia with a process-oriented focus. We provided a logic-based characterization of some well-known adaptive educational hypermedia systems: ELM-Art, Interbook, NetCoach, AHA!, and KBS Hyperbook, and where able to described them by means of (meta-)data about the document space, observation data (at runtime required data about user interaction, user feedback, etc.), output data, and the processing data - the adaptation algorithms. As a result, we were able to formulate a catalogue of adaptation algorithms in which the adaptation result could be judged in comparison to the overhead required for providing the input data (comprising data about the document space and observation data and runtime). This catalogue provides a basis-set for re-usable adaptation algorithms.

Our second goal, designing and realizing personalized e-Learning in the Semantic Web which allows the learners to customize the degree, method and coverage of personalization, is subject-matter of the present paper. Our first step towards achieving this goal was to develop a generic architecture and framework, which makes use of Semantic Web technologies in order to realize Personal Learning Object Readers. These Personal Learning Object Readers are on the one hand *Readers*, which mean that they display learning objects, and on the other hand *Personal Readers*, thus they provide personalized contextual information on the currently considered learning object, like recommendations about additional readings, exercises, more detailed information, alternative views, the learning objectives, the application where this learning content is relevant, etc. We have developed a framework for creating and maintaining such Personal Learning Object Readers. The driving principle of this framework is to expose all the different personalization functionalities as *services* which are orchestrated by some mediator service. The resulting personalized view on the learning object and it's context is finally determined by another group of services which take care on visualization and device-adaptation aspects. The next step to achieve our second goal is to create an interface component which enables the learners to *select and customize* personalization services. This is object of investigation of our ongoing work. Other approaches to personalized e-learning in the Semantic Web can be taken, e.g. focusing on reuse of content or courses (e.g. [11]), or focusing on metadata-based personalization (e.g [6,3]). Also portal-strategies have been applied for personalized e-Learning (see [4]). Our approach differs from the above mentioned approaches as we encapsulate personalization functionality into specific services, which can be plugged together by the learner.

3. The Personal Reader Framework: Service-based Personalization Functionality for the Semantic Web

The Personal Reader framework [9] provides an environment for designing, maintaining and running personalization services in the Semantic Web. The goal of the framework is to establish personalization functionality as services in a semantic web. In the run-time component of the framework, Personal Reader instances are generated by plugging one or several of these *personalization services* together. Each developed Reader consists of a browser for learning resources *the reader part*, and a side-bar or remote, which displays the results of the personalization services, e.g. individual recommendations for learning resources, contextual information, pointers to further learning resources, quizzes, examples, etc. *the personal part* (see Figure 2). This section describes the architecture of the Personal Reader framework, and discusses authoring of Personal Readers within our framework.

3.1. Architecture

The architecture of the Personal Reader framework (PRF) makes use of recent Semantic Web technologies for realizing a service-based environment for implementing and accessing personalization services. The core component of the PRF is the so-called *connector service* whose task is to pass requests and processing results between the user interface component and available personalization services, and to supply user profile information, and available metadata descriptions on learning objects, courses, etc. In this way, the connector service is the mediator between all services in the PRF.

Two different kinds of services - apart from the connector service - are used in the PRF: personalization services and visualization services. Each *personalization service* offers some adaptive functionality, e.g. recommends learning objects, points to more detailed information, quizzes, exercises, etc. personalization services are available to the PRF via a service registry using the WSDL (Web Service Description Language, [15]). Thus, service detection and invocation takes place via the connector service which ask the web service registry for available personalization services, and selects appropriate services based on the service descriptions available via the registry.

The task of the *visualization services* is to provide the user interface for the Personal Readers: interprete the results of the personalization services to the user, and create the actual interface with reader-part and personalization-part.

The basic implementation guideline in the Personal Reader framework is the following: Whenever a service has to communicate with other services, we use RDF (Resource Description Framework, [14]) for describing requests, processing results, and answers. This has the immediate advantage, that all components of the Personal Reader framework (visualization services or personalization services) can be independently developed, changed or substituted, as long as the interface protocol given in the RDF descriptions is respected. To make these RDF descriptions "understandable" for all services, they all externalize their meaning by referring to (one or several) ontologies. We have developed an ontology for describing adaptive functionality, the l3s-ontology[1]. Whenever a personalization service is implemented, the provided adaptation of this service is described with respect to this adaptation ontology, such that each visualization service can interprete the meaning of the adaptation, and can decide which presentation of the results should be used in accordance to the device that the user currently has, or the available bandwidth. This has the consequence, that local context adaptation (e.g. adaptation based on the capabilities of the user's device, bandwidth, environment, etc.) is not done by the personalization services, but by the visualization services. Figure 1 depicts the data flow in the PRF.

3.2. Authoring

Authoring is a very critical issue for successfully realizing adaptive educational hypermedia systems. As our aim in the Personal Reader framework is to support re-usability of personalization functionality, this is an especially important issue here. Recently, standards for annotating learning objects have been developed (cf. LOM [13] or IMS [12]). As a guideline for our work, we established the following rule:

[1] http://www.personal-reader.de/rdf/l3s.rdf

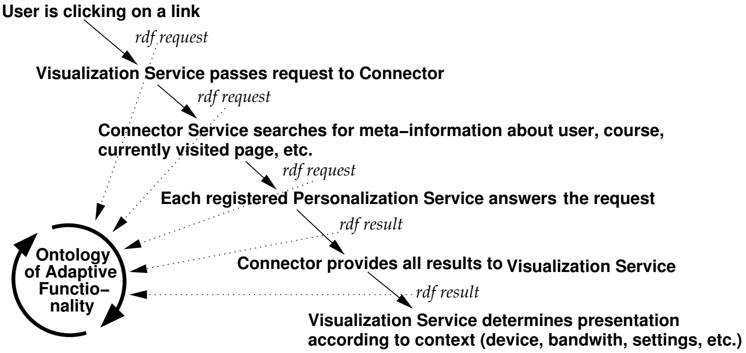

User is clicking on a link
rdf request
Visualization Service passes request to Connector
rdf request
Connector Service searches for meta–information about user, course, currently visited page, etc.
rdf request
Each registered Personalization Service answers the request
rdf result
Connector provides all results to Visualization Service
rdf result
Visualization Service determines presentation according to context (device, bandwith, settings, etc.)
Ontology of Adaptive Functio–nality

Figure 1. The communication flow in the Personal Reader framework: All communication is done via RDF-descriptions for requests and answers. The RDF descriptions are understood by the components via the *ontology of adaptive functionality*

> Learning Objects, course description, domain ontologies, and user profiles *must* be annotated according to existing standards (for details please refer to [8]). The flexibility must come from the personalization services which must be able to reason about these standard-annotated learning objects, course descriptions, etc.

This has an immediate consequence: We can implement personalization services which fulfill the same goal (e.g. providing a personal recommendations for some learning object), but which consider different aspects in the metadata. E.g. a personalization service can calculate recommendations based on the structure of the learning materials in some course and the user's navigation history, while another checks for keywords which describe the learning objectives of that learning objects and calculates recommendations based on relations in the corresponding domain ontology. Examples of such personalization services are given in Section 4.

The administration component of the Personal Reader framework provides an author interface for easily creating new instances of course-Readers: Course materials which are annotated according to LOM (or some subset of it), and which might in addition refer to some domain ontology, can immediately be used to create a new Personal Reader instance which offers all the personalization functionality which is - at runtime - available in the personalization services.

4. Realizing Personalization Services for e-Learning

This sections describes in more detail the realization of some selected personalization services: A service for recommending learning resources, and a service for enriching learning objects with the context in which they appear in some course.

4.1. Calculating Recommendations.

Individual recommendations for learning resources are calculated according to the current learning progress of the user, e. g. with respect to the current set of course materials. As described in Section 3.2, it is the task of the personalization services to realize strategies and algorithms which make use of standardized metadata annotations of learning objects, course descriptions, etc.

The first solution for realizing a *recommendation service* determines that a learning resource LO is recommended if the learner has studied at least one more general learning resource (UpperLevelLO), where "more general" is determined according to the course descriptions: :

```
FORALL LO, U learning_state(LO, U, recommended) <-
  EXISTS UpperLevelLO (upperlevel(LO, UpperLevelLO) AND
                        p_obs(UpperLevelLO, U, Learned) ).
```

Further personalization services can derive stronger recommendations than the previous one (e. g., if the user has studied *all* general learning resources), or less strong recommendations (e.g., if one or two of these haven't been studied so far), etc.

A different realization of a recommendation service can calculate its results with respect to the keywords describing the objectives of the learning object in some domain ontology. In particular, this is an appropriate strategy if a user is regarding course materials from different courses at the same time.

```
FORALL LO, U learning_state(LO, U, recommended) <-
  EXISTS C, C_DETAIL (concepts_of_LO(LO, C_DETAIL)
     AND detail_concepts(C, C_DETAIL) AND p_obs(C, U, Learned) ).
```

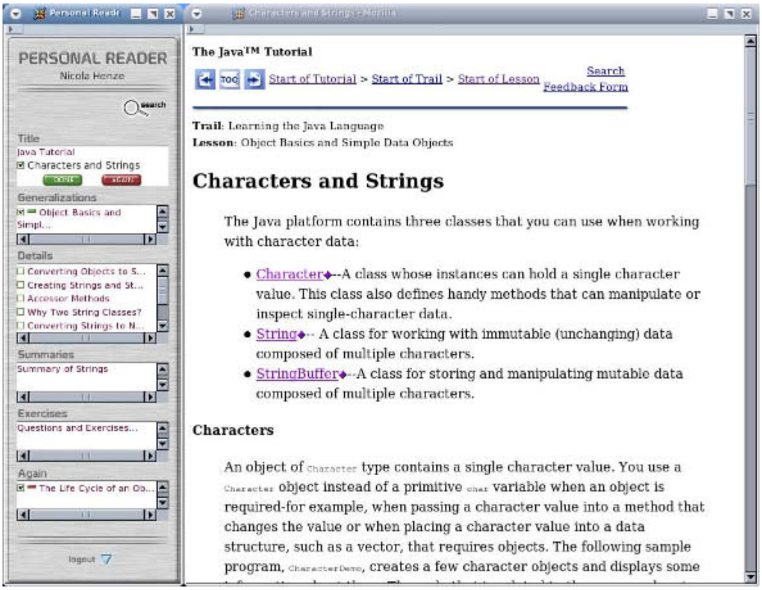
Comparing the above strategies for recommendation service we see that some of the recommendation services might provide better results as others - depending on the situation in which they are used. For example, a recommendation service, which reasons about the course structure will be more accurate than others, because it has more fine–grained information about the course and therefore on the learning process of a learner who is taking part in this course. But if the learner switches between several courses, recommendations based solely on the content of learning objects might provide better results. Overall, this yields to a configuration problem, in which we have to rate the different services which provide the same personalization functionality according to which data they used for processing, and in which situation they should been employed. We are currently exploring how we can solve this configuration problem with defeasible logics.

4.2. Course Viewer

For viewing learning objects which belong to some lecture, it is essential to show the learner the context of the learning objects: what is the general learning goal, what is this learning object about, and what are details that are related to this specific learning object. For example, a personalization service can follow the strategy to determining such details by following the course structure (if such a hierarchical structure like sections, subsections, etc. is given). Or it can use the key-concepts of the learning object and determine details with respect to the domain ontology.

The following rule applies the latter approach: Details for the currently regarded learning resource are determined by detail_learningobject(LO, LO_DETAIL) where LO and LO_Detail are learning resources, and where LO_DETAIL covers more specialized learning concepts which are determined with help of the domain ontology.

```
FORALL LO, LO_DETAIL detail_learningobject(LO, LO_DETAIL) <-
   EXISTS C, C_DETAIL(detail_concepts(C, C_DETAIL)
     AND concepts_of_LO(LO, C) AND concepts_of_LO(LO_DETAIL, C_DETAIL))
     AND learning_resource(LO_DETAIL) AND NOT unify(LO,LO_DETAIL).
```


Personal Reader
PERSONAL READER
Nicola Henze
search
Title
Java Tutorial
☒ Characters and Strings
Generalizations
☒ Object Basics and Simpl...
Details
□ Converting Objects to S...
□ Creating Strings and St...
□ Accessor Methods
□ Why Two String Classes?
□ Converting Strings to N...
Summaries
Summary of Strings
Exercises
Questions and Exercises...
Again
☒ The Life Cycle of an Ob...
logout
Characters and Strings - Mozilla
TOC
Start of Tutorial > Start of Trail > Start of Lesson
Search
Feedback Form

The Java™ Tutorial

Trail: Learning the Java Language
Lesson: Object Basics and Simple Data Objects

Characters and Strings

The Java platform contains three classes that you can use when working with character data:

- Character◆--A class whose instances can hold a single character value. This class also defines handy methods that can manipulate or inspect single-character data.
- String◆-- A class for working with immutable (unchanging) data composed of multiple characters.
- StringBuffer◆--A class for storing and manipulating mutable data composed of multiple characters.

Characters

An object of `Character` type contains a single character value. You use a `Character` object instead of a primitive `char` variable when an object is required-for example, when passing a character value into a method that changes the value or when placing a character value into a data structure, such as a vector, that requires objects. The following sample program, `CharacterDemo`, creates a few character objects and displays some

Figure 2. Screenshot of a Personal Reader for a e-Learning course on "Java Programming". The so far implemented Personal Readers are freely available at `www.personal-reader.de`.

4.3. Basic User Modeling

At the current state, the Personal Reader requires only few information about the user's characteristics. Thus, for our example we employed a very simple user model: This user model traces the users path in the learning environment and registers whenever the user has visited some learning resource. This simple user model is queried by all personalization services; updating the user model is task of the visualization services which provide the user interface and monitor user interactions.

4.4. Examples of Personal Learning Object Readers

Up to now, we have developed two Personal Learning Object Readers with our environment: A Personal Reader for learning the Java programming language (see the screenshot in figure 2), and a Personal Reader for learning about the Semantic Web. The Personal Reader for Java uses materials from the online version of the Sun Java Tutorial[2], while the one for learning about the Semantic Web uses materials of a course given at University of Hannover in summer 2004[3].

5. Conclusion and Future Work

This paper describes our approach for realizing personalized e-Learning in the Semantic Web. Our approach is driven by the goal of realizing a Plug & Play architecture for per-

[2] http://java.sun.com/docs/books/tutorial/

[3] http://www.kbs.uni-hannover.de/ henze/semweb04/skript/inhalt.xml

sonalized e-Learning which allows a learner to select, customize and combine personalization functionality. To achieve this goal, we have developed a framework for creating and maintaining personalization services, the *Personal Reader framework*. This framework provides an environment for accessing, invoking and combining personalization services, and contains a flexible, service-based infrastructure for visualizing adaptation outcomes, and for creating the user interface. Up to know, we have realized two Personal Readers (for the domains of Java programming and Semantic Web). Currently, we are implementing further personalization services, and are extending the user modeling component of the Personal Reader framework. Future work will include an improved way for combining personalization service, and for detecting and solving potential conflicts between the recommendations of these services.

References

[1] Ariadne: Alliance of remote instructional authoring and distributions networks for europe, 2001. http://ariadne.unil.ch/.

[2] Tim Berners-Lee, Jim Hendler, and Ora Lassila. The semantic web. *Scientific American*, May 2001.

[3] P. De Bra, A. Aerts, D. Smits, and N. Stash. AHA! version 2.0: More adaptation flexibility for authors. In *Proceedings of the AACE ELearn'2002 conference*, October 2002.

[4] P. Brusilovsky and H. Nijhawan. A framework for adaptive e-learning based on distributed re-usable learning activities. In *In Proceedings of World Conference on E-Learning, E-Learn 2002*, Montreal, Canada, 2002.

[5] Peter Brusilovsky. Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 11:87–110, 2001.

[6] Owen Conlan, Cord Hockemeyer, Vincent Wade, and Dietrich Albert. Metadata driven approaches to facilitate adaptivity in personalized elearning systems. *Journal of the Japanese Society for Information and Systems in Education*, 42:393–405, 2003.

[7] Edutella, 2001. http://edutella.jxta.org/.

[8] Nicola Henze, Peter Dolog, and Wolfgang Nejdl. Reasoning and ontologies for personalized e-learning. *ETS Journal Special Issue on Ontologies and Semantic Web for eLearning*, 2004. To appear.

[9] Nicola Henze and Matthias Kriesell. Personalization Functionality for the Semantic Web: Architectural Outline and First Sample Implementation. In *Proceedings of the 1st International Workshop on Engineering the Adaptive Web (EAW 2004)*, Eindhoven, The Netherlands, 2004.

[10] Nicola Henze and Wolfgang Nejdl. A logical characterization of adaptive educational hypermedia. *New Review of Hypermedia*, 10(1), 2004.

[11] Sebastien Iksal and Serge Garlatti. Adaptive web information systems: Architecture and methodology for resuing content. In *Proccedings of the 1st International Workshop on Engineering the Adaptive Web (EAW 2004)*, Eindhoven, The Netherlands, 2004.

[12] IMS: Standard for Learning Objects, 2002. http://www.imsglobal.org/.

[13] LOM: Draft Standard for Learning Object Metadata, 2002. http://ltsc.ieee.org/wg12/index.html.

[14] Resource Description Framework (RDF) Schema Specification 1.0, 2002. `http://www.w3.org/TR/rdf-schema`.

[15] WSDL: Web Services Description Language, version 2.0, August 2004. http://www.w3.org/TR/2004/WD-wsdl20-20040803/.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Making an Unintelligent Checker Smarter:

Creating Semantic Illusions from Syntactic Analyses

Kai Herrmann and Ulrich Hoppe
University of Duisburg-Essen, Germany.
{herrmann, hoppe}@collide.info

Abstract. In interactive learning environments based on visual representations the problem of checking the correctness of student solutions is much harder than with linear textual or numerical input. This paper presents a generic approach to providing learner feedback in such environments. The underlying checking mechanism analyzes system states against target constellations defined as sets of constraints about locations or connections of objects. A newly introduced analysis graph allows also the definition of procedural dependencies. An implementation of a feedback mechanism for traffic education in secondary schools is presented as an example.

Keywords. Pseudo Tutoring, Intelligent Tutoring, Error Analysis

1. Introduction: Modular Constraint Checking

Many interactive learning environments are based on visual representations which makes it much more difficult to check the correctness of student solutions than it is the case with linear textual or numerical input. In former articles [1,2], we have introduced a general approach to implement a checking mechanism for configuration problems of such visual languages essentially based on syntactic features of the underlying representation. Although this mechanism (called "modular constraint checking": MCC) is "unintelligent" in that it does not rely on semantic knowledge, it is capable of creating the illusion of having deep domain knowledge by providing highly contextual feedback to the learner.

First, we will briefly recapitulate the basic ideas behind the MCC: It is a state based (1.2) checking mechanism for visual languages (1.1), and capable of producing the impression or "illusion" of deep domain knowledge (1.3). Then, we describe new challenges for this approach (sec. 2) and how they are implemented (sec. 5). An example use case shows the advantages of the relatively light-weight form of checking in this approach as compared to more sophisticated (and more complicated!) systems (sec. 3).

1.1. Checking Visual Languages

Although there are some checking systems for dialog driven interfaces [3,4,5], there is a lack of systems which are able to check (and provide feedback for) visual languages in a more specific sense: A visual language consists of graphical objects the user can arbitrarily arrange on the screen. Values and positioning of the objects *together* form expressions in visual languages. There are two main problems in checking visual languages (compared with checking "regular", fixed interfaces):

- A checking mechanism for visual languages must be aware of the absolute and relative positions of objects on the screen and the connections between them. These facts are often as important as the values of the components itself.
- While it is simple to identify objects in fixed interfaces, where users only change the *value*, but not the *position* of objects, there is a problem in identifying objects in a visual language: Given, there are two objects (**x** and **y**) that represent the same concept in a visual language and differ only by their value. A user can, then, simply switch the values (and the position) from **x** to **y** and vice versa, so that **x** is now at the position **y** was before and has the value **y** had. *For the user* the objects now have changed their identity. But *for the system* they are still the old objects, but with changed values and positions. In such cases, the system must be able to handle objects according to the understanding of the user.

So, the MCC checking system uses especially information about location and connections of objects to identify them. Often a non-ambiguous identification is impossible, so the MCC checker has to deal with this fact. (See [2] for technical details.)

1.2. Checking States

When working with visual languages users typically modify objects following the direct manipulation paradigm. That means, e.g., moving an object to a certain position may include placing the object somewhere *near* the designated position in a first step, and in further steps, refining the position until the object ist placed *exactly* where it belongs to. Each single action of this sequence is not very expressive, nor is the sequence at a whole. Another user may use a completely different sequence of actions to move the object to the designated location, because there are literally thousands of ways to do so. Because (sequences of) single actions in the manipulation of visual languages is often not important, we do not observe *actions of users*, but *states of the system*. When observing a move operation, our system only recognizes the two states “object has reached the destination” and “object has not yet reached the destination”.

This approach differs from the approach of [6], which also describes a tutoring system that uses “weak” AI methods and programming by example. But Koedinger et al. examine user actions instead of states and build up a graph of (correct and faulty) user behavior, the *behavior graph* (BG). Each edge in this graph represents (one or more) user actions. Solving a task means doing actions which are marked as correct in the BG. If a user leaves these paths of correct actions, he or she gets error messages. The disadvantage of that approach is the fact that *all* possible actions users can execute while fulfilling a task must be inserted into the BG before. For visual languages, this is difficult, as pointed out before. Even integrating the logged actions of a big number of users into the graph (behavior recording, [7]) cannot solve this problem, because the number of possible sequences to solve a task is too big.

1.3. Semantic Illusion

To avoid the costs and complex problems of building checking systems which work with domain models [3], we focus on checking relatively low-level generic attributes of objects. We do not try to interpret these attributes on a domain level, but confine ourself to the analysis of connections between objects and their locations on the screen. Nevertheless, remaining on this lower level of interpretation we create feedback that appears to the user *as if* the checking system would possess deep domain knowledge. We call this *as if* behavior "semantic illusion". For each single case the system is prepared for, it is impossible to distinguish between such a "pseudo tutor" and a full sized intelligent tutoring system.[4] This approach releases us from building a domain model, and, by this, makes our system easily portable to new domains. Especially for the interaction with our learning environment Cool Modes [8], which is able to work with many visual languages from different domains, this is an advantage: The MCC checking system is able to work with all these languages with no or only very little porting effort.

2. New Challenges

Based on the concepts described in the last section, we are developing the following enhancements to the MCC system, which will be explained in detail in the sec. 5.

2.1. Supporting Non Experts in Extending the System

When developing new tutoring systems, a problem often mentioned is the fact that domain experts for learning scenarios (e.g. teachers) are normally *not* experts in AI programming. Thus, teachers are not able to build their own tutoring/checking system, because these computer related skills are necessary to build such a system.

With the enhancements of the MCC system we overcome the barrier between author and system designer [9], because things a system designer normally has to do on an *implementation level* at design time (writing code in a programming language) is now broken down to *configuration level* and can be done at use time by a domain expert. In this way, we enable a flexible transition from example authoring to system extension.

2.2. Aspect Handling

So far, the MCC system analyzes objects on the level of single attributes (e.g. the color of an object, or its x- and y-position on the screen). To make it easier for users to work with the MCC system, we have now added the concept of *aspects*. *Aspects* represent another type of constraints that implement higher-level concepts. Examples of *aspects* are:

- Absolute position of an object on the screen,
- relative position of two objects to each other,
- unique identification of objects,
- connections of an object to other objects.

If a user wants to observe one (or more) of these facets of an object, he or she does not have to deal with low-level parameters, but can simply select the suitable aspect(s), leaving the details to the system. It is easy to combine different aspects (e.g., unique identification and absolute position), and even mixing aspect constraints and other (lower level) attributes is possible. Additionally, users can make "snapshots" of given object constellations that hold information about one or more aspects of this constellation. This is like using a camera to make a photo of these objects. Such a "snapshot" can then be used as a target constellation for the checking system.

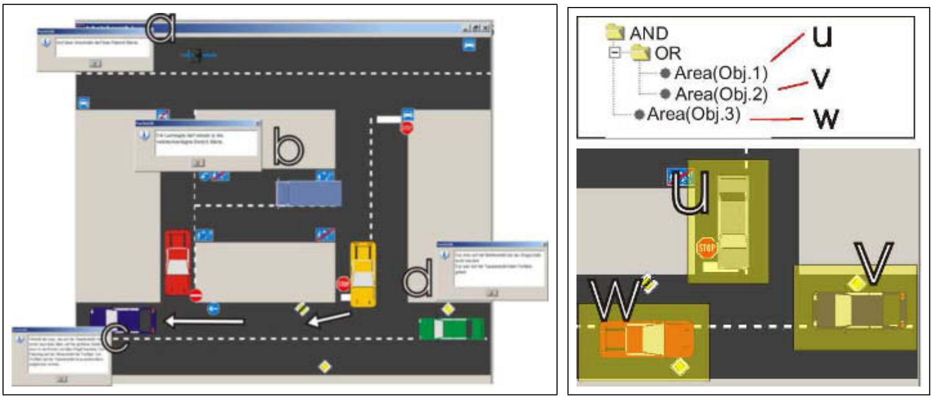


Figure 1. *Left side:* Picture of a complex traffic situation with message boxes generated by the MCC checking system. They describe four of about 20 situations that are observed by the MCC checking system in this scenario. In situation **a** and **b** a traffic participant is at a place where he is not allowed to be. In both cases a message box is shown that calls attention to that fact.The text box in situation **c** appears after the user has moved the car at the big horizontal street from the left of the crossing to the right. The message tells the user that the car in the one way street that touches the crossing from the top would have had the right of way, although the horizontal street is bigger. Situation **d** shows a combination of streets nearly identical to situation c. But now there are traffic signs that annul the right-before-left-rule. Here, the car on the horizontal road would have had the preference. *Right side:* At the top a condition tree that implements the situation shown at the bottom.

2.3. *Process Specification*

As mentioned above, we do not observe user actions, but system states. Nonetheless, often these states have to be reached in a certain chronological order. To be able to define such sequences of system states, we have now added a process model that represents sequential dependencies that are to be controlled at a given time. The use case described in section 3 uses an advanced feature of this process model that allows the definition of rules about system states that are active if given preconditions are fulfilled.

3. An Example Scenario: Traffic Education with Cool Modes

In the following, we describe an example use case for the MCC checking system from the domain of traffic education at primary schools. The scenario is realized by implementing an interactive traffic board, on which users can arrange streets, traffic signs and different kinds of vehicles. This interactive board is realized as a plug-in for the Cool Modes learning environment.[8] The MCC checking system is already integrated into Cool Modes. So, we can use it together with the traffic plug-in (as with any other plug-in) instantly, without further porting effort.

The left side of fig. 1 shows a scenario for the traffic plug-in for Cool Modes. You can see five streets with, at all, six crossings. Four cars, a truck and a bicycle drive through the streets. Various traffic signs rule the rights of way between the traffic participants.

The very existence of this setup makes teaching traffic education easier than with a blackboard. But the plug-in does not only show pictures, but it also “knows” the traffic rules that apply to the different locations of the map. The four text boxes in fig. 1 (left side) show messages that appear when the user violates a traffic rule while he or she moves a vehicle across the screen. So, the user can move vehicles across the streets of this example and explore all the things that might go wrong when driving a car. Every time he or she violates a rule, the system reports the error to the user. In addition, in many cases the user gets suggestions how he or she can avoid errors in the future.

4. Imagine Doing this with an *Intelligent* Checker...

In section 5 we will see how the MCC checking system implements checking the situations in the example scenario. But before let us consider the problems a checking system would have if it would try to solve these situations based on domain knowledge:

- All relevant traffic rules must be formalized and modelled.
- The system must be able to deal with inaccuracies, e.g. when the user places a car slightly beside a lane. So it must implement some kind of "fuzzy" recognition of situations.
- In the example in figure 1 the system seems to make guesses about the *reasons* of errors. So, an intelligent system must add heuristics to generate such tips for the user.

On the other hand, the big advantage of a knowledge based implementation of the traffic rules (and an important limitation of the MCC system) is that it would work with other street configurations as well, while the approach presented here restricts checking to one single given configuration. Using the "stupid" MCC approach, an author must build a new configuration file for each new street set up. But it is very questionable whether it would be worth doing the great effort of implementing an intelligent checker with a domain model for this traffic scenario, because scenarios like the one in figure 1 are complicated (and thus expensive) to model with a rule driven system. The implementation only pays off, if the resulting checker is used for many scenarios, and thus the cost is shared between the different applications. An ad-hoc-implementation by a teacher for the use at school next day can be done better and faster using the approach presented in this paper.

5. Solutions

In this section we will see how the MCC checking system produces the illusion "as if" it knows something about traffic rules. Also, we will explain the new features aspect handling and process specification (cf. section 2).

5.1. How to Specify a Target Constellation

Although a good part of the highway code is involved in the traffic example presented in the last section, nothing of this code is modelled for the checking facilities of the MCC system. Instead, just the parameters for the location and size of the objects are needed. The right side of fig. 1 shows how the (semantic, domain specific) traffic rules are broken down to a level of checking locations: The crossing in the figure involves concepts like STOP and right-of-way signs, in concurrence to the right-before-left rule. But the concrete situation can also be described with two simple sentences:

- If there is a car at **v** or **w**, the car at **u** is not allowed to drive. (This sentence is modelled by the condition tree at the top right side of figure 1 (right side).
- If there is a car at **v**, the car at **w** is not allowed to do a left turn.

There is no need to explain the highway code to the system, as long as it can check these two simple sentences. The system does this by simply checking, whether there is an (better: any) object at a specified screen location, highlighted in fig. 1 (right side). By this, the system can "recognize" domain-specific situations like right-of-way conditions without knowing anything about cars and streets.

Because the analysis only uses methods that are inherited from the super class of all visual components in Cool Modes (x and y coordinate), there is no need to adjust the checking system for dealing with traffic education. The support of this domain is for free. For other Cool Modes plug-ins it may be necessary to provide specialized forms of analysis. But even this specialized analysis methods can be added *at runtime* by configuration, not by implementation.

5.2. Aspect Handling

Using an older version of the MCC checking system [2], a user had to implement an examination of object locations by using low-level parameters like `x`, `y`, `width`, and `height`. He or she *can* still do so with the new system. But in most cases this is unnecessary. To provide a more practical, user oriented way of specifying target constellations, we added *aspects* to the MCC. An aspect is a new type of constraint that can be used instead of a bundle of (low-level) attributes to realize a higher level concept. E.g., the concept "absolute position on the screen" is implemented by combining the parameters `x`, `y`, `width`, and `height`. If a user wants to check the position of an object, he or she does not have to deal with low-level parameters, but can simply select the suitable aspect from a list, even without knowing which parameters in detail are substituted by this aspect.

The attributes forming the aspect "absolute position" are quite obvious. Less obvious are the attributes defining the aspect "identification", that is a collection of attributes that faces the problem of defining identity in visual languages, mentioned in section 1.1. This aspect does not comprise a fixed set of attributes, but different attributes, depending on the object that is to be identified.

To instantly produce a target constellation for a check, users can make *snapshots* of a given group of objects. While doing so, the system adapts (one or more) aspects to each member of a group of objects and adds the result of this operation to a constraint tree.

5.3. Sequences of Target Constellations

The MCC checking system has the ability to survey not only single target constellations, but also sequences of these. Going back to the traffic example in figure 1 (right side), we see that the correct handling of the right-of-way situations needs the analysis of *two different* situations:

- First, the system has to recognize that there is a situation that *may* cause a violation of the right-of-way rule. When the system recognizes such a situation, the rule is switched on.

- Second, the system must survey, if, with his or her next step, the user *actually breaks* the rule. Only in this case a feedback will be provided. If, on the other hand, the user resolves the situation correctly, the rule is switched off silently.

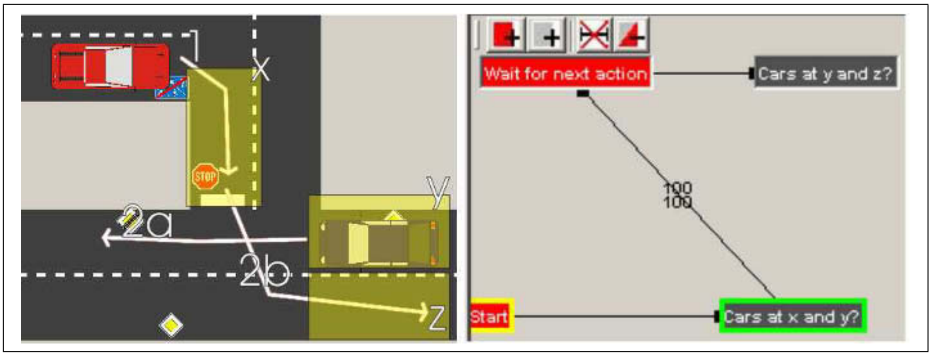


Figure 2. The right side of this figure shows a graph, in which the nodes on the right side (“Cars at...”) represent a target constellation. Also, each of these nodes can have an output associated with it. The graph realizes a simplified version of the right-of-way rule for the crossing at the left of this figure.
At the beginning, the “Start” node is active and surveys the first target constellation (cars at x and y). The target constellation is not fulfilled, and so nothing happens. After moving the car from the top to area x (1), the target constellation is fulfilled. The processor now follows the edge to the next node, which says "Wait for next action". Now the processor surveys the second target constellation (cars at y and z). If the user makes an error now and moves the car from area x to area z (2b) the second target constellation is fulfilled. There is an output connected with this configuration (not shown here) and the user will be informed that he or she made an error concerning the stop sign. Otherwise, if the user (correctly) moves the car from area y (2a), there will be no message. Neither the first, nor the second target constellation is fulfilled any longer (there is just a car left in area x), and so the processor starts again surveying the first target constellation only.

Fig. 2 shows in detail how this sequencing process works. In the use case described in section 3, about 20 rules like this are active simultaneously. Of course, the “chains” built by surveyed target constellations can be longer than shown in fig. 2. Here, there is just a precondition and a postcondition. As long as the precondition is fulfilled, the system surveys the postcondition.

The sequencing mechanism in the MCC checking system has the same function as the behavior graph in the CTAT environment of [4]. It connects points of interest throughout the checking process and gives them a consistent order. But while the behavior graph is restricted in the way that it only works with sequences of user actions that are defined before, the processor graph is more flexible and provides more freedom to the user: In the example in fig. 2 the user can do arbitrary actions; but every time he or she produces a situation matching the first target condition, the rule will switch to active state. Now, again, the user can do arbitrary actions, maybe at other areas of the map, with other cars at other crossings, the rule waits until a relevant state is reached and then reports an error or switches off silently. Compared with this, in any given situation, the behavior graph can only handle user actions which are provided for this concrete situation. Parallelism (user does something different first before continuing his or her actual task) and unexpected user behavior are much more complicated to handle with the behavior graph.

6. Conclusion

In this paper we presented the MCC system, a method to check visual language configuration problems without the use of deep domain knowledge and without “strong” AI methods. The MCC system is effective when feedback should be provided for a smaller number of problems in a given domain. Additionally, the system can be customized for new domains by *domain* (not AI) specialists. The MCC checker has been tested with configuration problems from various domains, e.g. models of mathematical proofs, petri nets, and context sensitive help systems. Although constraint satisfaction problems (CSPs) in general can have exponential complexity, the complexity of *average* MCC configuration files is usually more like $O(n^2)$, because most of the constraints are local. So, the system can also handle more complex cases without run time problems.

A limitation of the system is that an author has to create a new configuration file for each new case. The bigger the number of cases from a single domain, the more it is worthwhile to invest in the work of building a real ITS based on strong AI methods. But for a teacher who just wants to set up one or two situations for next day’s use, the MCC system is much better suited.

Currently, we are building a MCC checker to provide context sensitive help for a complex visual language concerning stochastic experiments. Another idea (not put into practice yet) is to use a MCC checker as an agent to move the cars in the use case described in this paper. The cars would then move across the traffic setting automatically, behaving in accordance with the highway code but without having any idea of it.

References

[1] K. Gassner, M. Jansen, A. Harrer, K. Herrmann, and U. Hoppe. Analysis methods for collaborative models and activities. In Proceedings of the CSCL 2003, pp. 369–377.

[2] K. Herrmann, U. Hoppe, and N. Pinkwart. A checking mechanism for visual language environments. In Proceedings of the AIED 2003, pp. 97–104.

[3] T. Murray. Authoring intelligent tutoring systems. *Int. Journal of AIEd*, 10:98–129, 1999.

[4] K. Koedinger, V. Aleven, and N. Heffernan. Essentials of cognitive modeling for instructional design: Rapid development of pseudo tutors. In *Proceedings of the ICLS*, 2004.

[5] Hot Potatoe. http://web.uvic.ca/hrd/halfbaked/, 2004.

[6] K. Koedinger, V. Aleven, N. Heffernan, B. McLaren, and M. Hockenberry. Opening the door to non-programmers: Authoring intelligent tutor behavior by demonstration. In *Proceedings of the ITS*, 2004.

[7] B. McLaren, K. Koedinger, M. Schneider, A. Harrer, and L. Bollen. Towards cognitive tutoring in a collaborative web-based environment. In Maristella Matera and Sara Comai, editors, *Engineering Advanced Web Applications*, Paramus, USA, 2004. Rinton Press.

[8] N. Pinkwart. A plug-in architecture for graph based collaborative modeling systems. In Proceedings of the AIED 2003, pp. 535–536.

[9] G. Fischer and E. Giaccardi. *End User Development*, chapter Meta-Design: A Framework for the Future of End-User Development. Kluwer Academic Publishers, 2004.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Iterative Evaluation of a Large-Scale, Intelligent Game for Language Learning

W. Lewis Johnson, Carole Beal
Center for Advanced Research in Technology for Education
USC / Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, CA 90292

Abstract. Educational content developers, including AIED developers, traditionally make a distinction between formative evaluation and summative evaluation of learning materials. Although the distinction is valid, it is inadequate for many AIED systems because they require multiple types of evaluation and multiple stages of evaluation. Developers of interactive intelligent learning environments must evaluate the effectiveness of the component technologies, the quality of the user interaction, and the potential of the program to achieve learning outcomes, in order to uncover problems prior to summative evaluation. Often these intermediate evaluations go unreported, so other developers cannot benefit from the lessons learned. This paper documents the iterative evaluation of the Tactical Language Training System, an interactive game for learning foreign language and culture. This project employs a highly iterative development and evaluation cycle. The courseware and software have already undergone six discrete stages of formative evaluation, and further formative evaluations are planned. The paper documents the evaluations that were taken at each stage, as well as the information obtained, and draws lessons that may be applicable to other AIED systems.

Introduction

Educational content developers conventionally draw a distinction between formative and summative evaluation of educational materials. Formative evaluation takes place during development; it seeks to understand strengths and amplify them, and understand weaknesses and mend them, before the educational materials are deployed. Summative evaluation is retrospective, to document concrete achievement [5]. Many view formative evaluation as something that should be kept internal to a project, and not published. This is due in part to the belief that formative evaluations need not involve learners. For example, Scriven [6] is frequently quoted as having said: “When the cook tastes the soup, that’s formative; when the guests taste the soup, that’s summative.”

Although the formative vs. summative distinction is useful, it does not provide much guidance to AIED developers. AIED systems frequently incorporate novel computational methods, realized in systems that must be usable by the target learners, and which are designed to achieve learning outcomes. These issues all warrant evaluation, and the “cooks” cannot answer the evaluation questions simply by “tasting the soup.” Yet one cannot use summative evaluation methods for this purpose either. Multiple evaluation questions need to be answered, which can involve multiple experiments, large numbers of subjects and large amounts of data. Meanwhile the system continues to be developed, so by the time the evaluation studies are complete they are no longer relevant to the system in its current form.

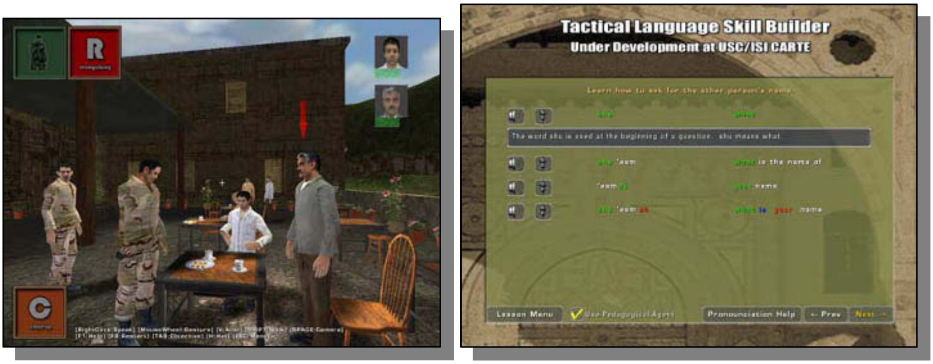
This paper documents the formative evaluation process to date for the Tactical Language Training System (TLTS). This project aims to create computer-based games, incorporating artificial intelligence technology, and each supporting approximately 80 hours of learning.

Given the effort required to create this much content, evaluation with learners could not wait until the summative stage. Instead, a highly iterative formative evaluation process was adopted, involving six discrete evaluation stages so far. Representative users were involved in nearly all stages. Each individual evaluation was small scale, but together they provide an accumulating body of evidence from which to predict that the completed system will meet its design objectives. The evaluation process has enabled the TLTS to evolve from an exploratory prototype to a practical training tool that is about to be deployed on a wide scale. These evaluation techniques may be relevant to other AIED projects that wish to make a smooth transition from the research laboratory to broad-based educational use.

1. Overview of the Tactical Language Training System

The Tactical Language Training System is designed to help people rapidly acquire basic spoken conversation skills, particularly in languages that few foreigners learn because they are considered to be very difficult. Each language training package is designed to give people enough knowledge of language and culture to carry out specific tasks in a foreign country, such as introducing yourself, obtaining directions, and arranging meetings with local officials. The curriculum and software design are focused on the necessary skills for the target tasks, i.e., it has a strong task-based focus [3]. The current curricula focus on the needs of military personnel engaged in civil affairs missions; however the same method could be applied to any language course that focuses on communication skills for specific situations. Two training courses are being developed so far: Tactical Levantine Arabic, for the Arabic dialect spoken in Lebanon and surrounding countries, and Tactical Iraqi, for the Iraqi Arabic dialect.

The TLTS includes the following main components [8]. The Mission Game (Figure 1, left side) is an interactive story-based 3D game where learners practice carrying out the mission. Here the player's character, at middle left, is introducing himself to a Lebanese man in a café. The player is accompanied by an aide character (far left), who can offer suggestions if the player gets stuck. The Skill Builder (Figure 1, right) is a set of interactive exercises focused on the target skills, in which learners practice saying words and phrases, listening to and responding to sample utterances. A virtual tutor evaluates the learner's speech and gives feedback that provides encouragement and attempts to overcome learner negative affectivity [7]. A speech-enabled Arcade Game gives learners further practice opportunities. Finally, there is a hypertext glossary can show the vocabulary in each lesson, the grammatical structure of the phrases being learned, and explains the rules of grammar that apply to each utterance.



Learn how to ask for the other person's name.

The word shu is used at the beginning of a question. shu means what

Figure 1. Views of the Tactical Language Training System

2. Evaluation Issues for the TLTS

The stated goal of the TLTS project is to enable learners with a wide range of aptitudes to acquire basic conversational proficiency in the target tasks, in a difficult language such as Arabic, in as little as eighty hours of time on the computer. We believe that achieving this goal requires a combination of curriculum innovations and new and previously untested technologies. This raises a host of evaluation issues and difficulties. It is hard to find existing courses into which TLTS can be inserted for evaluation purposes, because the TLTS curriculum and target population differ greatly from that of a typical Arabic language course. Most Arabic courses place heavy emphasis on reading and writing Modern Standard Arabic, and are designed for high-aptitude learners. The TLTS Arabic courseware focuses on spoken Arabic dialects, and is designed to cater to a wide range of learners with limited aptitude or motivational difficulties. The TLTS employs an innovative combination of gaming and intelligent tutoring technologies; this method needed to be evaluated for effectiveness. It incorporates novel speech recognition [11], pedagogical agent [7] and autonomous agent technologies [14], whose performance must be tested. Because of the large content development commitment, content must be evaluated as it is developed in order to correct design problems as early as possible. It is not even obvious how much content is needed for 80 hours of interaction.

Then once the content is developed, additional evaluation questions come up. Standard language proficiency assessments are not well suited for evaluating TLTS learning outcomes. The most relevant assessment is the Oral Proficiency Interview (OPI), in which a trained interviewer engages the learner in progressively more complex dialog in the foreign language. Since TLTS learners apply language to specific tasks, their score on an OPI may depend on the topic that is the focus of the conversation. So-called task-based approaches to assessment [3] may be relevant, but as Bachman [1] notes, it is difficult to draw reliable conclusions about learner proficiency solely on the basis of task-based assessments. Thus TLTS faces a similar problem to other intelligent tutoring systems such as the PUMP Algebra Tutor [9]: new assessment instruments must be developed in order to evaluate skills that the learning environment focuses on. Finally, we need to know what components of the TLTS contribute to learning effectiveness; there are multiple components which may have synergistic effects.

3. Evaluating the Initial Concept

The project began in April of 2003, and focused initially on Levantine Arabic, mainly because Lebanese speakers and data sets are readily available in the United States. Very early on, an interactive PowerPoint mockup of the intended user interaction was developed and presented to prospective stakeholders. This was followed by simple prototypes of the Mission Game and Skill Builder. The Mission Game prototype was created as a “mod” of the Unreal Tournament 2003 game, using the GameBots extension for artificially intelligent characters (http://www.planetunreal.com/gamebots/). It allowed a learner to enter the virtual café shown in Figure 1, engage in a conversation with a character to get directions to the local leader’s house, and then follow those directions toward that house. The Skill Builder prototype was implemented in ToolBook, with enough lessons to cover the vocabulary needed for the first scene of the Mission Game, although not all lessons were integrated with the speech recognizer.

This prototype then was delivered to the Department of Foreign Languages at the US Military Academy (USMA) for formative evaluation. The USMA was a good choice for assisting the evaluation because they are interested in new technologies for language learning, and they have an extensive Arabic language program that provides strong training in spoken Arabic. They assigned an experienced Arabic student (Cadet Ian Strand) to go through the lesson materials, try to carry out the mission in the MPE, and report on the potential value of the software for learning. CDT Strand was not a truly representative user, since he already knew Arabic and had a high language aptitude. However he proved to be an ideal evaluator at this stage—he was able to complete the lessons and mission even though the lessons were incomplete, and was able to evaluate the courseware from a hypothetical novice's perspective.

An alternative approach at this stage could have been to test the system in a Wizard-of-Oz experiment. Although Wizard-of-Oz experiments can be valuable for early evaluation [13], they have one clear disadvantage—they keep the prototype in the laboratory, under the control of an experimenter. By instead creating a self-contained prototype with limited functionality, we obtained early external validation of our approach.

4. Adding Functionality, Testing Usability

Several months of further development and internal testing followed. The decentralized architecture of the initial prototypes was replaced with an integrated multi-process architecture [8]. Further improvements were made to the speech recognizer, and the lesson and game content were progressively extended. Then in April 2004 we conducted the next formative evaluation with non-project members.

Seven learners participated in this study. Most were people in our laboratory who had some awareness of the project; however none of them had been involved in the development of the TLTS. Although all had some foreign language training, none of them knew any Arabic. All were experienced computer game players. They were thus examples of people who ultimately should benefit from TLTS, although not truly representative of the diversity of learners that TLTS was designed to support.

The purpose of this test was to evaluate the usability and functionality of the system from a user's perspective. Each subject was introduced to the system by an experimenter, and was videotaped as they spent a one-hour session with the software, using a simplified thinking aloud protocol [13]. Afterwards the experimenter carried out a semi-structured interview, asking the subject about their impressions of different parts of the system.

No major usability problems were reported, and none appeared on the videotape. The subjects asserted that they felt the approach was *much* better than classroom instruction. Subjects who had failed to learn very much in their previous foreign language classes were convinced that they would be able to learn successfully using this approach. The subjects also felt that the game and lesson components supported each other, that if they had spent more time in the lessons it would help their performance in the game.

At the same time, a number of problems emerged, both in the instructional design and in the use of the underlying technology. The pronunciation evaluation in the Skill Builder was too stringent for beginners; this created the impression that the primary learning objective was pronunciation instead of communication. The feedback of the pedagogical agent was repetitive and sometimes incorrect. Because we had designed the pedagogical agent to act human-like, instances of repetitive, non-human-like behaviour were especially glaring. Some subjects were unsure of where to go in the game and what to do. There was also a general reluctance to play the game, for fear that it would be too difficult. Once they got to the game, they had difficulty applying the knowledge that they had acquired in the Skill Builder.

These evaluations led to system improvements. The library of tactics employed by the pedagogical agent was greatly extended, pronunciation accuracy threshold was lowered, and speech recognition performance was improved. More simulated conversation exercises were added to the Skill Builder, to facilitate transfer of knowledge to the Mission Game. An introductory tutorial was created for the Mission Game, in order to help learners get started.

5. A Comparative Test with Representative Users

A more extensive test was then conducted in July of 2004 with representative users. It was structured to provide preliminary evidence as to whether the software design promotes learning, and identify what parts of the software are most important in promoting learning.

The following is a brief overview of this study, which is described in more detail in [2]. Twenty-one soldiers at Ft. Bragg, North Carolina, were recruited for the study. The subjects were divided in four groups, in a 2x2 design. Two groups got both the Skill Builder and Mission Game, two got just the Skill Builder. Two groups got a version of the Skill Builder with pronunciation feedback, two groups got no pronunciation feedback. This enabled us to start to assess the role that tutorial feedback and gameplay might have on learning outcomes. Due to the limited availability of test computers each group only had six hours to work with the software over the course of a week, so learning gains were expected to be limited.

The group that worked with the complete system rated it as most helpful, considered it to be superior to classroom instruction, and in fact considered it to be comparable to one-on-one tutoring. On the other hand, the group that got tutorial feedback without the Mission Game scored highest on the post-test. It appeared that combination of performance feedback and motivational feedback provided by the virtual tutor helped to keep the learners engaged and focused on learning. Some reported that the found the human-like responses to be enjoyable and "cool". Apparently the shortcomings that the earlier study had identified in the tutorial feedback had been corrected.

Another important lesson from this study was how to overcome learners' reluctance to enter the Mission Game. We found that if the experimenter introduced them directly to the game and encouraged them to try saying hello to one of the characters there, they got engaged, and were more confident to try it. With the assistance of the virtual tutor, many were able to complete the initial scenario in the first session.

Improvement was found to be needed in the Mission Game and the post-test. The Mission Game was not able to recognize the full range of relevant utterances that subjects were learning in the Skill Builder. This and the fact that there are only a limited range of possible outcomes of the game when played in beginner mode gave learners the impression that they simply needed to memorize certain phrases to get through the game. After the first day the subjects showed up with printed cheat-sheets that they had created, so they could even avoid memorization. We concluded that the game would need to support more variability in order to be effective. On the evaluation side, we are concerned that the post-test that we used was based on the Skill Builder content, so that it did not really test the skills that learners should be acquiring in the game, namely to carry on a conversation.

We subsequently made improvements to the Mission Game language model and interaction so that there was more variability in game play. We also came up with a way to make the post-test focus more on conversational proficiency: to use the Mission Game as an assessment vehicle. If the virtual tutor in the game is replaced by another character who knows no Arabic, the learner is then forced to perform the task unassisted. If they can do this, it demonstrates that they have mastered the necessary skills, at least in that context. To make this approach viable, it would be necessary to log the user's interaction with the software. Therefore logging capabilities were added to enable further analysis of learner performance.

6. A Longer-Term Test with Representative Users

Once these and other improvements were made to the system, and more content was added, another test was scheduled at Ft. Bragg, in October, 2004. This time the focus was on the following questions. (1) How quickly do learners go through the material? (2) How proficient are they when they complete the material? (3) How do the subjects' attitudes and motivation affect performance, and vice versa? Question 1 was posed to extrapolate from the work completed so far and estimate how much additional content would be required to complete an

80-hour course. Question 2 was posed to assess progress toward achieving task-based conversational proficiency. In particular, we wanted to assess whether our proposed approach of using the Mission Game as an assessment tool was workable. Question 3 was of interest because we hypothesized that the benefits of TLTS result in part from improved learner motivation, both from the game play and from the tutorial feedback.

For this study, rather than break the subjects into groups, we assembled just one group of six subjects, and monitored them through three solid days of work with the program followed by a post-test. They were also soldiers, with a range of different aptitudes and proficiencies, although being members of the US Army Special Forces their intelligence was greater than that of the average soldier. Their ages ranged from 20 to 46 years, and all had some foreign language background; one even had some basic training in Modern Standard Arabic. Not surprisingly, all subjects in this study performed better than in the previous study, and performance was particularly good on vocabulary recognition and recall, understanding conversations, and simulated participation in conversations. They were also able to perform well in the Mission Game when employed as an assessment tool. They made better use of the Mission Game, and did not rely on cheat sheets this time. Overall, the utility of the Mission Game was much more apparent this time.

Although most of the subjects did well, two had particular difficulties. One was the oldest subject, who repeatedly indicated that he was preparing to retire from the military soon and had little interest in learning a difficult language that he would never use. The other subject expressed a high degree of anxiety about language learning, and that anxiety did not significantly abate over the course of the study.

Meanwhile, other problems surfaced. The new content that had been introduced in time for this evaluation still had some errors, and the underlying software had some bugs that impeded usability. The basic problem was that once the evaluation was scheduled, and subjects were accrued, it was impossible to postpone the test to perform further debugging. Given the choice of carrying out the test with a buggy version of the program and cancelling it altogether, the better choice was to go ahead with the evaluation and make the best of it.

Another problem came up during analysis of the results: the log files that were collected proved to be very difficult to use. Questions that were easy to pose, e.g., "How long did each subject take on average per utterance in the final MPE test scene?" in fact proved to be difficult to answer. The log files that the TLTS generated had not been constructed in such a way as to facilitate the kinds of analyses that we subsequently wanted to perform. In a sense we relearned the lesson that other researchers have identified regarding interaction logs [10]: that log analysis is more that data collection, and attention must be paid both to the design of the logging facility and to the tools that operate on the resulting logs. Fortunately our iterative evaluation approach enabled us to learn this lesson quickly and correct the situation before subsequent evaluations.

7. Formative Evaluation of Tactical Iraqi

After having responded to the lessons learned from the previous test and corrected some of the errors in the Levantine Arabic content, we then temporarily put Levantine Arabic aside and focused on developing new content for Iraqi Arabic. There was a political reason for this (the desire to do something to improve the political situation in Iraq), a technical reason (to see if the TLTS was generalizable to new languages), and a pedagogical reason (to see if our understanding of how to develop content for the TLTS had progressed to the point where we could develop new courses quickly). Iraqi Arabic is substantially different from Levantine Arabic, and Iraqi cultural norms different from Lebanese cultural norms. Nevertheless our technical and pedagogical progress were such that by January 2005 we had a version of

Tactical Iraqi ready for external formative evaluation that was already better developed than any of the versions of Tactical Levantine Arabic that have been developed to date.

During January we sent out invitations to US military units to send personnel to our laboratory to attend a seminar on the installation and use of Tactical Iraqi, and to take the software back with them to let other members of their units use. Three units sent representatives. It was made clear to them that Tactical Iraqi was still undergoing formative evaluation, and that they had critical roles to play in support of the formative evaluation. During the seminar the participants spent substantial amounts of time using the software and gave us their feedback; meanwhile their interaction logs and speech recordings were collected and used to further train the speech recognizer and identify and correct program errors. All participants were enthusiastic about the program, and two of the three installed it at their home sites and solicited the assistance of other members of their unit in beta testing. Logs from these interactions were sent back to CARTE for further analysis.

Periodic testing continued through the spring of 2005, and two more training seminars were held. A US Air Force officer stationed in Los Angeles volunteered to pilot test the entire course developed to date in May. This will be followed in late May by a complete learning evaluation of the content developed to date, at Camp Pendleton, California. Fifty US Marines will complete the Tactical Iraqi course over a two week period, and then complete a post test. All interaction data will be logged and analyzed. Camp Pendleton staff will informally compare the learning gains from this course against learning from their existing classroom-based four-week Arabic course.

During this test we will employ new and improved assessment instruments. Participants will complete a pre-test, a periodic instrument to assess their attitudes toward learning, and a post-test questionnaire. The previous learning assessment post-test has been integrated into the TLTS, so that the same mechanism for collecting log files can also be used to collect post-test results. We have created a new test scene in the Mission Game in which the learner must perform a similar task, but in a slightly different context. This will help determine whether the skills learned in the game are transferable. We will also employ trained oral proficiency interviewers assess the learning gains, so that we can compare these results against the ones obtained within the program.

Although this upcoming evaluation is for higher stakes, it is still formative. The content for Tactical Iraqi is not yet complete. Nevertheless, it is expected that the Marines will make decisions about whether to incorporate Tactical Iraqi into their language program. Content development for the current version of Tactical Iraqi will end in June 2005, and summative evaluations at West Point and elsewhere are planned for the fall of 2005.

4. Summary

This article has described the formative evaluation process that was applied in the development of the Tactical Language Training System. The following is a summary of some of the key lessons learned that may apply to other AIED systems of similar scale and complexity. Interactive mock-ups and working prototypes should be developed as early as possible. Initial evaluations should if possible involve selected individuals who are not themselves target users but can offer a target user's perspective and are able to tolerate gaps in the prototype. Preliminary assessments of usability and user impressions should be conducted early if possible, and repeated if necessary, in order to identify problems before they have an impact on learning outcomes. In a complex learning environment with multiple components, multiple small-scale evaluations may be required until all components prove to be ready for use. Design requirements are likely to change based on lessons learned from earlier formative evaluations, which in turn call for further formative evaluation to validate them.

Mostow [10] has observed that careful evaluation can be onerous, and for this reason researchers tend to avoid it or delay it until the end of a project. An iterative evaluation method is infeasible if it involves a series of onerous evaluation steps. Instead, this paper illustrates an approach where each evaluation is kept small, in terms of numbers of subjects, time on task, and/or depth of evaluation. The individual studies may yield less in the way of statistically significant results than large-scale evaluations do, but the benefit is that evaluation can be tightly coupled into the development process, yielding a system that is more likely to achieve the desired learning outcomes when it is complete. The experience gained in the formative pilot evaluations will moreover make it easier to measure those outcomes.

Acknowledgments

This project is part of the DARWARS Training Superiority Program of the Defense Advanced Research Projects Agency. The authors wish to acknowledge the contributions of the members of the Tactical Language Team. They also wish to thank the people at the US Military Academy, Special Operations Foreign Language Office, 4th Psychological Operations Group, Joint Readiness Training Center, 3rd Armored Cavalry Division, 7th Army Training Command, and Marine Corps Expeditionary Warfare School for their assistance in the evaluations described here.

References

[1] Bachman, L.F. (2002). Some reflections on task-based language performance assessment. *Language Testing* 19(3), 461-484.

[2] Beal, C., Johnson, W.L., Dabrowski, R., & Wu, S., (2005). Individualized feedback and simulation-based practice in the Tactical Language Training System: An experimental evaluation. AIED 2005. IOS Press.

[3] Bygate, M., Skeehan, P., & Swain, M. (2001). *Researching pedagogic tasks: Second language learning, teaching, and testing*. Harlow, England: Longman.

[4] Corbett, A.T., Koedinger, K.R. & Hadler, W.S. (2002). Cognitive Tutors: From research classroom to all classrooms. In P. Goodman (Ed.): *Technology enhanced learning: Opportunities for change*. Mahwah, NJ: Lawrence Erlbaum Associates.

[5] The Center for Effective Teaching and Learning, University of Texas at El Paso. Formative and summative evaluation. http://cetal.utep.edu/resources/portfolios/form-sum.htm.

[6] Scriven, 1991, cited in "Summative vs. Formative Evaluation", http://jan.ucc.nau.edu/edtech/etc667/proposal/evaluation/summative_vs._formative.htm

[7] Johnson, W.L., Wu, S., & Nouhi, Y. (2004). Socially intelligent pronunciation feedback for second language learning. ITS '04 Workshop on Social and Emotional Intelligence in Learning Environments.

[8] Johnson, W.L., Vilhjálmsson, H., & Marsella, S. (2004). The DARWARS Tactical Language Training System. Proceedings of I/ITSEC 2004.

[9] Koedinger, K.R., Anderson, J.R., Hadley, W.M., & Mark, M.A. (1997). Intelligent tutoring goes to school in the big city. *IJAIED*, 8, 30-43.

[10] Mostow, J. (2004). Evaluation purposes, excuses, and methods: Experience from a Reading Tutor that listens. *Interactive Literacy Education: Facilitating Literacy Environments Through Technology*, C. K. Kinzer, L. Verhoeven, ed., Erlbaum Publishers, Mahwah, NJ.

[11] Mote, N., Johnson, W.L., Sethy, A., Silva, J., Narayanan, S. (2004). Tactical language detection and modeling of learning speech errors: The case of Arabic tactical language training for American English speakers. InSTIL/ICALL Symposium, Venice, Italy.

[12] Nielsen, J. (1994). Guerrilla HCI: Using discount usability engineering to penetrate the intimidation barrier. http://www.useit.com/papers/guerrilla_hci.html

[13] Rizzo, P., Lee, H., Shaw, E., Johnson, W.L,, Wang, N., & Mayer, R. (2005). A semi-automated Wizard of Oz interface for modeling tutorial strategies. UM'05.

[14] Si, M. & Marsella, S. (2005). Thespian: Using multiagent fitting to craft interactive drama. AAMAS 2005.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Cross-Cultural Evaluation of Politeness in Tactics for Pedagogical Agents

W. Lewis Johnson[1], Richard E. Mayer[2], Elisabeth André[3], Matthias Rehm[3]
[1]*USC / Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, CA 90292*
[2]*University of California, Santa Barbara*
[3]*University of Augsburg, Institute for Computer Science, Eichleitnerstr. 30, Germany*

Abstract. Politeness may play a role in tutorial interaction, including promoting learner motivation and avoiding negative affect. Politeness theory can account for this as a means of mitigating the face threats arising in tutorial situations. It further provides a way of accounting for differences in politeness in different cultures. Research in social aspects of human-computer interaction predict that similar phenomena will arise when a computer tutor interacts with learners, i.e., they should exhibit politeness, and the degree of politeness may be culturally dependent.

To test this hypothesis, a series of experiments was conducted. First, American students were asked to rate the politeness of possible messages delivered by a computer tutor. The ratings were consistent with the conversational politeness hypothesis, although they depended upon the level of computer literacy of the subjects. Then, the materials were translated into German, in two versions: a polite version, using the formal pronoun Sie, and a familiar version, using the informal pronoun Du. German students were asked to rate these messages. Ratings of the German students were highly consistent with the ratings given by the American subjects, and the same pattern was found across both pronoun forms.

1. Introduction

Animated pedagogical agents are capable of rich multimodal interactions with learners [6, 14]. They exploit people's natural tendency relate to interactive computer systems as social actors [16], to respond to them as if they have human qualities such as personality and empathy. In particular, pedagogical agents are able to perform affective and motivational scaffolding [2, 4, 9]. Educational researchers have increasingly called attention to the role of affect and motivation in learning [13, 17] and the role of expert tutoring in promoting affective and motivational states that are conducive to learning [11, 12]. Pedagogical agents are being developed that emulate motivational tutoring tactics, and they can positively affect learner attitudes, motivational state, and learning gains [18].

We use the politeness theory of Brown and Levinson [3] as a starting point for modelling motivational tactics. Politeness theory provides a general framework for analyzing dialog in social situations, and in particular the ways in which speakers mitigate face threats. When human tutors interact with learners they constantly risk threatening the learner's face, by showing disapproval or taking control away from the learner. They can also enhance learner face by showing approval and respect for the learner's choices. This in turn can have an impact on the learner's attitude and motivation. Johnson et al. [10] have developed a model for characterizing tutorial dialog moves in terms of the amount of face threat redress they exhibit, and implemented it in a tutorial tactic generator that can vary the manner in which a tutorial dialog move is realized depending upon the degree of attention paid to the learner's face and motivational state.

An interesting aspect of Brown and Levinson's theory is that it applies to all languages and cultures. Every language has a similar set of methods for mitigating face threat; however, not all cultures ascribe equal importance to each type of face threat. Using politeness theory as a framework, it is possible to create tutorial tactics in multiple languages and compare them to assess their impact in different cultures.

This paper presents a study that performs just such a comparison. German subjects evaluated the degree of face threat mitigation implied by a range of tutorial tactics for a pedagogical agent. These ratings were compared against similar ratings by American subjects of pedagogical agent tactics in English. The ratings by the subjects were in very close agreement. Use of formal vs. informal pronouns, a cardinal indicator of formality in German, did not have a significant effect on ratings of face threat mitigation. These results have implications for efforts to adapt pedagogical agents for other languages and cultures, or to create multilingual pedagogical agents (e.g., [8]).

2. Background: Politeness Theory and Tutorial Dialog

An earlier study analyzed the dialog moves made by a human tutor working with learners on a computer-based learning environment for industrial engineering [7]. It was found that the tutor very rarely gave the learners direct instructions as to what to do. Instead, advice was phrased indirectly in the form of questions, suggestions, hints, and proposals. Often the advice was phrased as a proposal of what the learner and tutor could do jointly (e.g., "So why don't we go back to the tutorial factory?"), when in reality the learner was carrying out all of the actions. Overall, tutorial advice was found to fall into one of eight categories: (1) direct commands (e.g., "Click the ENTER button"), (2) indirect suggestions (e.g., "They are asking you to go back and maybe change it"), (3) requests, (4) actions expressed as the tutor's goals (e.g., "Run your factory, that's what I'd do"), (5) actions as shared goals, (6) questions, (7) suggestions of student goals ("e.g., "you will probably want to look at the work centres"), and (8) Socratic hints (e.g., "Well, think about what you did.").

Brown & Levinson's politeness theory provides a way to account for these indirect tutorial dialog moves. According to politeness theory, all social actors have *face wants*: the desire for *positive face* (being approved of by others) and the desire for *negative face* (being unimpeded by others). Many conversational exchanges between people, (e.g., offers, requests, commands) potentially threaten positive face, negative face, or both. To avoid this, speakers employ various types of face threat mitigation strategies to reduce the impact on face. Strategies identified by Brown and Levinson include positive politeness (emphasizing approval of the hearer), negative politeness (emphasizing the hearer's freedom of action, e.g., via a suggestion) and off-record statements (indirect statements that imply that an action is needed). The eight categories listed above fit naturally as subcategories of Brown and Levinson's taxonomy, and can be understood as addressing the learner's positive face, negative face, or both. In this corpus positive face is most often manifested by shared goals (the willingness to engage in shared activity with someone implies respect for that person's contributions). We hypothesize that tutors adjust their modes of address with learners not just to mitigate face threat, but also to enhance the learners' sense of being approved of and free to make their own choices. These in turn can have an influence on the learners' self-confidence, and these factors have been found by researchers on motivation (e.g. [12]) to have an impact on learner motivation.

Based on this analysis, Johnson and colleagues [11] developed a tutorial dialog generator that automatically selects an appropriate form for a tutorial dialog move, based on the social distance between the tutor and the learner, the social power of the tutor over the learner, the degree of influence the tutor wishes to have on the learner's motivational state, the type of face threatening action, and the degree of face threat mitigation afforded by each type of tutorial dialog move. The dialog generator utilizes a library of tutorial tactics, each of which is annotated according to the amount of redress that tactic gives to the learner's positive face and negative face. Once each tactic is annotated in terms of negative and positive face, the generator can choose appropriate tactics automatically.

To make this scheme work, it is necessary to obtain appropriate positive politeness and negative positive politeness ratings for each tactic. These ratings were obtained using an experimental method described in [13]. Two groups of instances of each of the eight tactic categories were constructed (see appendix). One set, the A group, consisted of recommendations to click the ENTER button on a keyboard. The B group consisted of suggestions to employ the quadratic formula to solve an equation. Two different types of advice were given in case the task context influences the degree of face threat implied by a particular suggestion. These advice messages were then presented to 47 experimental subjects at the University of California, Santa Barbara (UCSB), who were told to evaluate them as possible messages given by computer tutor. Each message was rated according to the degree to which it expressed respect for the user's choices (negative politeness) and a feeling of working with the user (positive politeness). The main findings were as follows:

- With this experimental instrument, subjects ascribed degrees of positive and negative politeness with a high degree of consistency;
- The rankings of the ratings were consistent with the rankings proposed by Brown and Levinson, suggesting that the subjects ascribed politeness to the computer tutor as if it were a social actor;
- The task context did not have a significant effect on the politeness ratings;
- Ratings of politeness *did* depend upon the amount of computer experience of the subjects—experienced computer users were more tolerant impolite tutor messages than novice computer users were.

Based upon these findings, it was concluded that politeness theory could be validly applied to dialog with a pedagogical agent, and that the average ratings for each type of tactic obtained from the study could be used to calibrate the tutorial tactic generator, possibly adjusting for the level of computer experience of the user.

3. Experimental Evaluation of Politeness in German

Having successfully applied to politeness theory to the choice of tutorial tactics in English, we then considered the question of whether it might equally apply to tutorial tactics in German. Politeness theory is claimed by Brown and Levinson to apply to dialog in all languages and cultures; however not all cultures attribute the same degree of face threat to a given face threatening act. We therefore attempted to replicate the UCSB study in German. We anticipated that the ratings given by German subjects might differ from the American ratings for any of the following reasons:

- Politeness theory might not apply cross-culturally to human-computer interaction as it does to human-human interaction;
- Certain face threats might be regarded as more serious in one culture than in the other;
- Human tutors in Germany might have different power or social distance relationships with human students, affecting the amount of face threat that learners tolerate;
- Translating the messages into German might introduce cultural issues that are absent in English and yet have an impact on perceived politeness.

The participants for the German experiments were 83 students from Augsburg University. Thirty-nine students were recruited from the Philosophy department while 44 students were recruited from the Computer Science department. One subject indicated using a computer 1 to 5 hours per week, 11 indicated using a computer 5 to 10 hours per week, 26 indicated using a computer 10 to 20 hours per week, and 45 indicated using a computer more then 20 hours per week. The mean age of the subjects was 22.8 years (SD=1.997). There were 37 women and 46 men. Seventy-eight of the 83 students reported German as their native language.

For the German experiment, we devised a German version of the original English questionnaire. We tried to find translations that closely matched the original English documents, but nevertheless sounded natural to native speakers of German. During the translation, the question arose of how to translate the English "you". There are different ways of saying "you" in German depending on the degree of formality. In German, the more familiar "Du" is used when talking to close friends, relatives or children, while people tend to use the more formal "Sie" when talking to adults they do not know very well or to people that have a high status. Whether to use "Sie" or "Du" may constitute a difficult problem both for native speakers of German and foreigners. On the one hand, the "Du" address form might be considered as impolite or even abusive. On the other hand, switching to the "Sie" address form may be interpreted as a sign that the interlocutor wishes to maintain distance. A German waiter in the pub that is mostly frequented by young people is in a dilemma when she has to serve somebody of an older age. Some customers might consider the "Du" as disrespectful. Other might be irritated by the "Sie" since it makes them aware of the fact that they belong to an older age group. Similar dilemmas may occur in the academic context. Most German professors would use "Sie" when addressing undergraduates, but "Du" is common as well.

Since address forms are an important means to convey in-group membership (see also [3]), we expected that the use of "Sie" or "Du" might have an impact on the students' perception of politeness. In particular, we assumed that the students might perceive an utterance as more cooperative if the "Du" is used (positive politeness). Furthermore, the students might feel under higher pressure to perform a task if the teacher conveys more authority (negative politeness).

To investigate these questions, we decided to divide the subjects into two groups. Thirty-seven students were presented with the more formal "Sie" version and 46 students were presented with the more confidential "Du" version of the questionnaire. That is, the variable "address form" was manipulated between subjects while comparisons concerning types of statements were within subject comparisons.

Do the two kinds of politeness rating correspond for the English and the German version?

Table 1 gives the mean ratings for each of the 16 sentences for the English and the German experiment on the rating scale for negative and positive politeness. Items were rated on a scale from 1 (least polite) to 7 (most polite). The items are listed in order of negative/positive politeness for the US condition. As in the US experiment, the most impolite statements are direct commands and commands attributed to the machine whereas the most polite statements are guarded suggestions and "we" constructions that indicate a common goal.

For set B, there are just two permutations between two neighbour positions (B1 ↔ B2, B6 ↔ B7) in the case of positive politeness. In the case of negative politeness the order of the statements of set B completely coincide. For set A, the order of the statements differs to a higher degree. In particular, item A5 got a much lower rating for negative politeness in Germany than in the US. As a reason, we indicate that the utterance "Drücken wir die ENTER Taste" (Let's click the ENTER button.) sounds rather patronizing in German which might have evoke the feeling in the students that the agent does not respect their freedom. This patronising impression engendered by the first person plural is not unique to German; for example, in English adults sometimes use this form when giving commands to children (e.g., "OK, Johnnie, let's go to bed now"). Nevertheless, the effect was obviously stronger for the German version, but interestingly only occurred for negative politeness. Both the American and the German subjects gave A5 the highest rating in terms of positive politeness.

Mean Ratings for Neg. Politeness for the Experiments Conducted in the US and in Germany

	A1	A2	A3	A4	A5	A6	A7	A8	B1	B2	B5	B4	B3	B8	B6	B7
US	1.75	2.72	2.89	3.17	3.34	4.28	4.51	5.85	1.79	2.75	3.26	3.32	3.79	4.11	4.70	4.83
D	1.42	2.70	2.65	3.70	1.93	4.35	4.06	5.49	1.43	2.10	3.31	3.76	4.08	4.17	4.60	5.39

Mean Ratings for Pos. Politeness for the Experiments Conducted in the US and in Germany

	A1	A2	A4	A3	A6	A8	A7	A5	B2	B1	B4	B8	B3	B6	B7	B5
US	2.53	2.94	3.32	3.85	4.09	4.11	4.83	5.17	3.06	3.09	4.04	4.43	4.79	4.89	4.95	5.26
D	3.04	2.87	3.98	3.28	4.72	4.83	4.48	4.87	2.45	2.41	4.27	4.27	5.04	5.23	5.20	5.66

Table 1: Comparison of the Experimental Results Obtained in the US and in Germany

Overall, the Pearson correlation between the US and German ratings of positive politeness for the 16 statements is r = .926 which is highly significant (p < .001). The correlation for US and German ratings of negative politeness for the 16 statements is r = .893 which is highly significant (p < .001) as well. This means that we can conclude that German and American users responded to the politeness level of our statements in the same way.

An analysis of variance conducted on the 8 items revealed that the ratings differed significantly from each other for negative politeness (F(7,574)=100.6022, p <.001 for set A, F(7,574)=98.8674, p<.001 for set B) and for positive politeness (F(7,574)=21.8328, p <.001 for set A, F(7,574)=51.3999, p <.001 for set B).

Do the two forms of the sentences correspond in the German version?

As in the US experiment, we analyzed whether the statements in set A conveyed the same politeness tone as the corresponding statements in set B. To accomplish this task, we computed Pearson correlations among the ratings of the 83 students on each pair of corresponding items (e.g., A1 and B1, A2 and B2, etc.) on each scale. Among the items on the first rating scale, only A1 and B1, A2 and B2, A4 and B4 as well as A6 and B6 correlated significantly at the .01 level. Among the items on the second rating scale, A1 and B1, A2 and B2, A4 and B4, A5 and B5, and A6 and B6 correlated significantly at the .01 level and A3 and B3 at the .05 level. There was no such strong correlation between A8 and B8 and A7 and B7 on any of the two scales. Overall, the students found the utterance „Möchten Sie die ENTER Taste drücken?" (Do you want to click the ENTER button?) more polite (on both scales) than „Haben Sie die Quadratformel verwendet, um diese Gleichung zu lösen?" (Did you use the quadratic formula to solve this equation). Furthermore, the utterance „Sie könnten die Quadratformel verwenden, um diese Gleichung zu lösen." (You could use the quadratic formula to solve this equation.) was perceived as more polite (on both scales) than „Sie möchten wohl die ENTER Taste drücken." (You may want to click the ENTER button). Since the direct translation of the English sentence sounded rather unusual, we decided to add the discourse particle "wohl" (well). In connection with "möchten" (want), "wohl" is, however, frequently be used to signal the addressee that she will not be able to perform the intended action. We assume that a more neutral wording "möchten wahrscheinlich" (probably want) instead of "möchten wohl" (well want) would have led to different results.

Does the address form "Du" or "Sie" in the German experiment make any difference?

Table 2 gives the mean rating for the 16 statements for negative and positive politeness in the "Du" and "Sie" conditions. The statements are listed in order of negative and positive politeness respectively. As you can see, the order of the sentences of set A and B does not differ drastically for the "Du" and the "Sie" version.

Mean Ratings for Negative Politeness

	A1	A5	A3	A2	A4	A7	A6	A8	B1	B2	B5	B4	B3	B8	B6	B7
Du	1.43	2.11	2.70	2.72	3.46	4.04	4.33	5.64	1.50	2.04	3.48	3.78	3.87	4.41	4.70	5.28
Sie	1.41	1.70	2.59	2.68	4.00	4.08	4.38	5.32	1.35	2.16	3.11	3.73	4.35	3.86	4.49	5.51

Mean Ratings for Positive Politeness

	A1	A2	A3	A4	A7	A6	A8	A5	B2	B1	B4	B8	B3	B6	B7	B5
Du	3.22	3.09	3.39	4.07	4.24	4.70	4.76	5.07	2.61	3.50	4.13	4.30	5.13	5.15	5.20	5.65
Sie	2.81	2.59	3.14	3.86	4.78	4.76	4.92	4.62	2.24	3.30	4.43	4.22	4.92	5.32	5.11	5.68

Table 2: Comparison of the Experimental Results for the "Du" and "Sie" Conditions

Overall, the Pearson correlation between the "Du" and the "Sie" version for negative politeness is r = .974 which is highly significant (p < .001). The correlation between Du and Sie forms for positive politeness is r = .968 which also is very strong (p < .001). The experiment clearly shows that the use of the address form did not influence the subjects' perception of politeness. Since the students were not given detailed information on the situational context, they obviously assumed a setting which justified the used address form. That is, the choice of an appropriate address form ensured a basic level of politeness, but did not combine additively with other conversational tactics.

4. Related Work

There has been a significant amount of research on universal and culture-specific aspects of politeness behaviours. Most noteworthy is the work by House who performed a series of contrastive German-English discourse analyses over the past twenty years, see [5] for a list of references. Among other things, she observed that Germans tend to be more direct, and more self-referenced, and resort less frequently to using verbal routines. While House focused on the analysis of spoken or written discourse, we were primarily interested in ranking communication tactics derived from a corpus of tutorial dialogues according to their perceived level of politeness. Hardly any work has addressed the cultural dimension of politeness behaviours in the context of man-machine communication so far.

Our work is closely related to the work of Porayska-Pomsta [15] analyzing politeness in instructional dialogs in the United States and Poland. Porayska-Pomsta also observes close similarities between the role of politeness in American tutorial dialogs and Polish classroom dialogs. However the two corpora that she studied were quite different in nature: one is text-based chat and the other is in-class dialog. It is therefore difficult to make the same kinds of quantitative comparisons between the two data sets that we have made here.

Alexandris and Fotinea [1] investigate the role of discourse particles as politeness markers to inform the design of a Greek Speech Technology application for the tourist domain. They performed a study in which evaluators had to rank different variations of written dialogues according to their perceived degree of naturalness and acceptability. The study revealed that dialogues in Modern Greek with discourse particles indicating positive politeness are perceived as friendlier and more natural while dialogues without any discourse particles or discourse particles fulfilling other functions were perceived as unnatural. The authors regard these findings as cultural-specific elements of the Greek language.

5. Conclusions

These studies have demonstrated that politeness theory applies equally to tutorial dialog tactics in English and in German, applied by pedagogical agents, as evaluated by university students in the United States and Germany. Politeness ratings are remarkably similar between the two languages and cultures. The "Du"/"Sie" distinction, which can be an important indicator of social standing in German society, does not have a significant influence on perceived politeness. There are some slight differences in judgments of politeness in individual cases, in part because direct translations are not always possible and the best equivalent translations sometimes connote a somewhat different degree of politeness. Nevertheless, the degree of correlations between American and German ratings is quite high. Obviously, the eight categories of commands retrieved from the US corpus are common in German tutorial dialogues as well. It would appear that tutorial tactics falling into these classes can be translated fairly freely between the German and American educational contexts, although one has to be careful to consider that possibility that individual tactics may have different politeness connotations in the other language.

These results are further evidence for the contention that developers of intelligent tutors should take into account the possibility that learners will relate to the tutors as if they were social actors.

Acknowledgments

This work was funded by the National Science Foundation under Grant No. 0121330 and BaCaTec. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Alexandris, C., & Fotinea, S. (2003). Discourse Particles: Indicators of Positive and Non-Positive Politeness in the Discourse Structure of Dialog Systems for Modern Greek. In: *SDV - Sprache und Datenverarbeitung*, Jahrgang 28, Heft 1, pp. 22-33, 2004.

[2] Bickmore, T. (2003). *Relational agents: Effecting change through human-computer relationships*. Ph.D. thesis, Massachusetts Institute of Technologgy.

[3] Brown, P., & Levinson, S. C. (1987). *Politeness: Some universals in language use*. New York: Cambridge University Press.

[4] Conati, C. & Zhao, X. (2004). Building and evaluating an intelligent pedagogical agent to improve the effectiveness of an educational game. Proceedings of IUI'04. New York: ACM Press.

[5] House, J. (2000). Understanding Misunderstanding: A Pragmatic Discourse Approach to Analysing Mismanaged Rapport in Talk Across Cultures. In: Spencer-Oatey, H. (Eds.). *Culturally Speaking: Managing Rapport Through Talk Across Cultures* . London: Cassell Academic.

[6] Johnson, W.L., Rickel, J., & Lester, J. (2000). Animated pedagogical agents: Face to face interaction in interactive learning environments. *IJAIED* 11:47-78.

[7] Johnson. W.L. (2003). Interaction tactics for socially intelligent pedagogical agents. *Proc. of the Int'l Conf. on Intelligent User Interfaces*, 251-253. New York: ACM Press, 2003.

[8] Johnson, W.L., LaBore, C., & Chiu, J. (2004). A pedagogical agent for psychosocial intervention on a handheld computer. AAAI Fall Symposium on Health Dialog Systems.

[9] Johnson, W. L., Rizzo, P. (2004). Politeness in tutorial dialogs: "Run the factory, that's what I'd do." *Proc. of the 7th International Conference on Intelligent Tutoring Systems*. Berlin: Springer.

[10] Johnson, W.L., Rizzo, P., Bosma, W., Kole, S., Ghijsen, M., & van Welbergen, H. (2004). Generating socially appropriate tutorial dialog. *Proceedings of ADS '04*. Berlin: Springer.

[11] Johnson, W.L., Wu, S., & Nouhi, Y. (2004). Socially intelligent pronunciation feedback for second language learning. ITS '04 Workshop on Social and Emotional Intelligence in Learning Environments.

[12] Lepper, M. R., Woolverton, M., Mumme, D., & Gurtner, J. (1993). Motivational techniques of expert human tutors: Lessons for the design of computer-based tutors. In S. P. Lajoie and S. J. Derry (Eds.), *Computers as cognitive tools* (pp. 75-105). Hillsdale, NJ: Erlbaum.

[13] Mayer, R.E., Johnson, W.L, Shaw, E., & Sandhu, S. (2005). Constructing Computer-Based Tutors that are Socially Sensitive: Politeness in Educational Software. Paper presented at the annual conference of the American Educational Research Association. Montreal, Canada.

[14] Moreno, R. (in press). Multimedia learning with animated pedagogical agents. In R. E. Mayer (Ed.), *Cambridge handbook of multimedia learning*. New York: Cambridge University Press.

[15] Porayska-Pomska, K. (2003). *The influence of situational context of language production: Modeling teachers' corrective responses*. Ph.D. thesis, Univesity of Edinburgh.

[16] Reeves, B., & Nass, C. (1996). *The media equation*. New York: Cambridge University Press.

[17] Sansone, C., and Harackiewicz, J. M. (2000). *Intrinsic and extrinsic motivation: The search for optimal motivation and performance*. San Diego: Academic Press.

[18] Wang, N., Johnson, W.L., Rizzo, P., Shaw,E., & Mayer, R. (2005). Experimental evaluation of polite interaction tactics for pedagogical agents. Proceedings of IUI '05. New York: ACM Press.

Appendix

A1 Click the ENTER button. / Drücken Sie die ENTER Taste.
A2 The system is asking you to click the ENTER button. / Das System bittet Sie, die ENTER Taste zu drücken.
A3 I would like you to click the ENTER button. / Ich hätte gerne, dass Sie die ENTER Taste drücken.
A4 I would now click the ENTER button. / Ich würde nun die ENTER Taste drücken.
A5 Let's click the ENTER button. / Drücken wir die ENTER Taste.
A6 And what about the ENTER button? /Und wie wäre es mit dem Drücken der ENTER Taste?
A7 You may want to click the ENTER button. / Sie möchten wohl die ENTER Taste drücken.
A8 Do you want to click the ENTER button? / Möchten Sie die ENTER Taste drücken?

B1 Now use the quadratic formula to solve this equation. / Nun verwenden Sie die Quadratformel, um diese Gleichung zu lösen.
B2 The machine wants you to use the quadratic equation. / Die Maschine möchte, dass Sie die Quadratformel verwenden, um diese Gleichung zu lösen.
B3 I suggest that you use the quadratic formula to solve this equation. / Ich schlage vor, dass Sie die Quadratformel verwenden, um diese Gleichung zu lösen.
B4 I would use the quadratic formula to solve this equation. / Ich würde die Quadratformel verwenden, um diese Gleichung zu lösen.
B5 We should use the quadratic formula to solve this equation. / Wir sollten die Quadratformel verwenden, um diese Gleichung zu lösen.
B6 What about using the quadratic formula to solve this equation? / Wie wäre es, wenn Sie die Quadratformel verwenden würden, um diese Gleichung zu lösen?
B7 You could use the quadratic formula to solve this equation. / Sie könnten die Quadratformel verwenden, um diese Gleichung zu lösen.
B8 Did you use the quadratic formula to solve this equation? / Haben Sie die Quadratformel verwendet, um diese Gleichung zu lösen?

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Serious Games for Language Learning: How Much Game, How Much AI?

W. Lewis Johnson, Hannes Vilhjalmsson, & Stacy Marsella
Center for Advanced Research in Technology for Education
USC / Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, CA 90292

Abstract. Modern computer games show potential not just for engaging and entertaining users, but also in promoting learning. Game designers employ a range of techniques to promote long-term user engagement and motivation. These techniques are increasingly being employed in so-called *serious games*, games that have non-entertainment purposes such as education or training. Although such games share the goal of AIED of promoting deep learner engagement with subject matter, the techniques employed are very different. Can AIED technologies complement and enhance serious game design techniques, or does good serious game design render AIED techniques superfluous? This paper explores these questions in the context of the Tactical Language Training System (TLTS), a program that supports rapid acquisition of foreign language and cultural skills. The TLTS combines game design principles and game development tools with learner modelling, pedagogical agents, and pedagogical dramas. Learners carry out missions in a simulated game world, interacting with non-player characters. A virtual aide assists the learners if they run into difficulties, and gives performance feedback in the context of preparatory exercises. Artificial intelligence plays a key role in controlling the behaviour of the non-player characters in the game; intelligent tutoring provides supplementary scaffolding.

Introduction

In the early days of intelligent tutoring system (ITS) research, intelligent tutors were conceived of not just as aids for academic problem solving, but as supports for interactive games. For example, Sleeman and Brown's seminal book, *Intelligent Tutoring Systems*, included two papers on tutors that interacted with learners in the context of games: the WEST tutor [1] and the Wumpus tutor [5]. It was recognized that games can be a powerful vehicle for learning, and that artificial intelligence could amplify the learning outcomes of games, e.g., by scaffolding novice game players or by reinforcing the concepts underlying game play.

Fast forward to 2005. Computer games have become a huge industry, a pastime that most college students engage in [12]. In their striving for commercial success, game developers have come up with a set of design principles that promote deep, persistent engagement, as well as learning [17]. Education researchers are now seeking to understand these principles, so that they can understand how to make education more effective [4]. There is increasing interest in *serious games*, programs that obey solid game design principles but whose purpose is other than to entertain, e.g., to educate or train [20]. Meanwhile, with a few exceptions (e.g., [2, 6]), very little current work in AI in education focuses on games.

This paper examines the question of what role artificial intelligence should play in serious games, in order to promote learning. The artificial intelligence techniques used must support the learning-promoting features of the game, otherwise they may be superfluous or even counterproductive. These issues are discussed in the context of the Tactical Language Training System (TLTS), a serious game for learning foreign language and culture.

1. Overview of the Tactical Language Training System

The language courses delivered using the TLTS have a strong task-based focus; they give people enough knowledge of language and culture to enable them to carry out particular tasks in a foreign country, such as introducing yourself, obtaining directions, and meeting with local officials. The current curricula address the needs of military personnel, however the same method could be applied to any course that focuses on the skills needed to cope with specific situations, e.g., booking hotel rooms or meeting with business clients. Two training courses have been developed so far: Tactical Levantine Arabic, for the Arabic dialect spoken in the Levant, and Tactical Iraqi, for Iraqi Arabic Dialect.

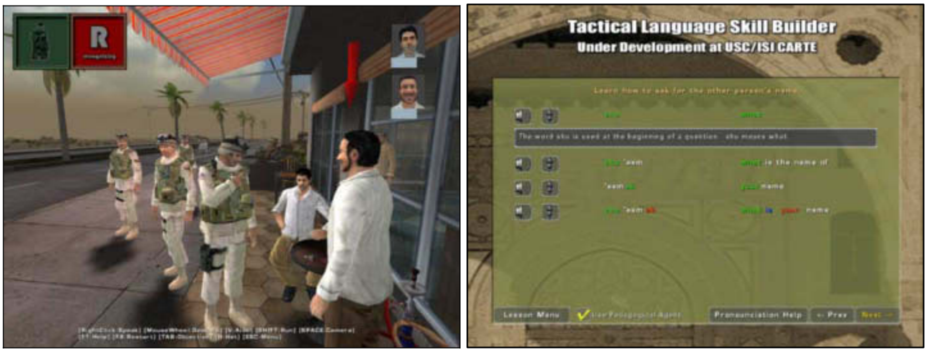


Figure 1. Views of the Tactical Language Training System

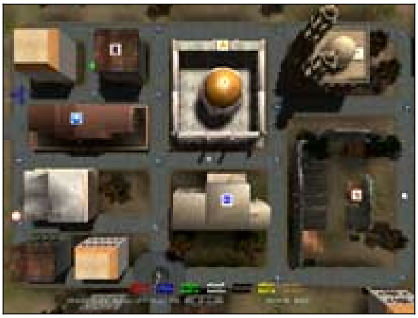


Figure 2. Arcade Game in the TLTS

The TLTS includes the following main components [8]. The Mission Game (Figure 1, left side) is an interactive story-based 3D game where learners practice carrying out the mission. Here the player's character, center, is introducing himself to an Iraqi man in a café, so that he can ask him where the local leader might be found. The player is accompanied by an aide character (middle left), who can offer suggestions of what to do if the player gets stuck. The Skill Builder (Figure 1, right) is a set of interactive exercises focused on the target skills and tasks, in which learners practice saying words and phrases, and engaging in simple conversations. A virtual tutor evaluates the learner's speech and gives feedback on errors, while providing encouragement and attempting to overcome learner negative affectivity [10]. A speech-enabled Arcade Game gives learners further practice in speaking words and phrases

(Figure 2). Finally, there is an adaptive hypertext glossary that shows the vocabulary in each lesson, and explains the grammatical structure of the phrases being learned.

The TLTS has been evaluated multiple times with representative learners, through an iterative formative evaluation process [11]. The evaluations provide evidence that the game format motivates learners who otherwise would be reluctant to study a difficult language such as Arabic. A significant amount of content is being developed, which by July 2005 should be able to support around 80 hours of interaction for Iraqi Arabic and somewhat less for Levantine Arabic. Multiple military training centers have volunteered to serve as test sites.

2. Using AI Design to Support Game Design

The premise of the serious game approach to learning is that well designed games promote learner states that are conducive to learning. Serious game developers adhere to a number of common design principles that tend to yield desirable interaction modes and learner states [4, 17]. Some of these principles are commonplace in AIED systems, particularly those that employ a goal-based-scenario approach [19]; others are less common, and may appear new to AIED developers. Game AI can play a critical role in implementing these principles. Game AI is a major research area in its own right, which goes beyond the scope of this paper (see [13] for an overview). In educational serious games, the challenges are to make sure the game AI supports educational objectives, and to introduce other educational AI functions as needed without compromising game design principles, in order maximize learning.

2.1. Gameplay

According to Prensky, one of the foremost characteristics of good games is good gameplay. "*Gameplay is all the activities and strategies game designers employ to get and keep the player engaged and motivated to complete each level and an entire game.*" [18] Good gameplay does not come from the game graphics, but from the continual decision making and action that engages the learner and keeps him or her motivated to continue.

There are two aspects of gameplay: engaging users moment by moment, and relating current game actions to future objectives. In good moment-by-moment gameplay, each action or decision tends to naturally lead to the next action or decision, putting the player in a psychological state of flow [3]. Moment-by-moment gameplay is realized in the Mission Game as follows. The actions in the Mission Game that relate to the target tasks (namely, face-to-face communication) are embedded in a larger sequence of navigation, exploration, and information gathering activities that learners engage in as they carry out their mission. When the learner is engaged in a conversation with a nonplayer character, there is a give and take between the characters; the nonplayer characters respond both verbally and nonverbally to the learner's utterances, and may take initiative in the dialog. In the Arcade Game, there is a constant flow of action and reaction between the user's actions (issuing spoken Arabic commands to navigate through the game level and pick up objects) and the game's response (moving the game character as indicated by the spoken commands, scoring points for correctly uttered phrases and collected items, and immediately placing new objects in the game level). In the Mission Game, orientation toward future objectives occurs as the learner develops rapport with the local people, and obtains information from them relevant to the mission. In the Arcade Game this orientation occurs as learners seek to increase their overall game score and progress to the next game level.

One way that AI facilitates gameplay in the TLTS is by promoting rapid interaction with nonplayer characters. Speech recognition in the game contexts is designed to rapidly and robustly classify the intended meaning of each learner utterance, in a manner that reasonably tolerant of learner errors, at least as much as human native speakers would be [9]. Natural language processing is employed to generate possible dialog variants that a learner might attempt to say during the game, but only at authoring time, to reduce the amount of game-time processing required on user input. The PsychSim package is then used to generate each character's responses to the learner's actions [21]. Pedagogical objectives are realized in PsychSim using an interactive pedagogical drama approach, by making sure that the nonplayer characters respond to aspects of learner communication that are pedagogically important (e.g., appropriate use of polite gesture and language).

On the other hand, the common use of AI in intelligent tutoring systems, to provide tutorial scaffolding, is carefully restricted in the TLTS. We avoid interrupting the gameplay with critiques of learner language. Such critiques are reserved for Skill Builder lessons and after-action review of learner performance.

2.2. Feedback

Good games provide users with feedback on their actions, so that they know how well they are doing and can seek to improve their performance. This has obvious relevance to serious games that motivate learners to improve their skills.

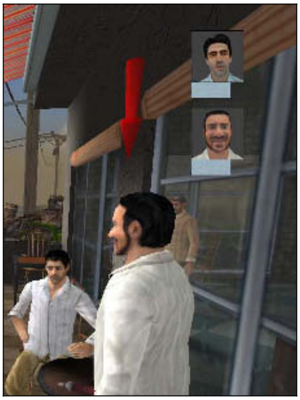


Figure 3. Close-up of the trust meters in the Mission Game

As we conducted formative evaluations of the TLTS, we frequently saw a need to improve feedback, and developed new feedback methods in response. For example, when learners carry out actions in the Mission Game that develop rapport with the local people (e.g., greet them and carry out proper introductions), they want to know if they are making progress. Some cues that people rely on in real life, such as the facial expressions of the people they are talking to, are not readily available in the game engine underlying TLTS (namely, Unreal Tournament 2003). We therefore developed an augmented view of the non-player characters' mental state, called a trust meter, shown in the upper right of Figure 3. The size of the grey bar under each character image grows and shrinks dynamically depending upon the current degree of trust that character has for the player. Note that this lessens the need for intelligent coaching on the subject of establishing trust, since learners can recognize when their actions are failing to establish trust.

2.3. Affordances

Another feature of good games is their simple, well-defined interfaces, designed to support the interaction between the user and the game. Even in games that attempt to create very realistic 3-D virtual worlds, designers will augment that reality in various ways to provide the user with "perceived affordances" [16], in essence cues that suggest or guide user actions. For example, in Figure 3 there is a red arrow above the head of one of the characters that informs the user about which character in the virtual world is engaging them in the conversation. More generally, the Mission Game uses icons and highlighting of the screen to help regulate the dialog turn-taking between the learner and the characters. Although this augmented reality diverges from strict realism both in terms of the rendering of the scene and the mechanisms used to regulate dialog turn-taking in real-life, they better serve the goal of maintaining a fluid interaction between the learner and the non-player characters. Again, effective use of affordances lessens the need for intelligent coaching to advise learners on what actions to take.

2.4. Challenge

An important aspect of game design is ensuring that users experience a proper level of challenge. Gee argues that the user experience should be "pleasantly frustrating:" a challenge for the player, but not an insurmountable one [4]. The role of challenge in promoting intrinsic motivation is not limited to games, but has been noted by motivation researchers as relevant to all learning activities [14].

The TLTS is configurable to adjust the level of challenge of play. When beginners play in the Mission Game, they receive assists in the form of subtitles showing what the Arab characters as saying, both in transliteration and in English translation. Also, each Mission Game scene can be played at two levels of difficulty, either Novice or Experienced. At the Novice level the Arab characters are relatively tolerant of cultural gaffes, such as failing to show proper respect or failing to make proper introductions. At the Experienced level the Arab characters become suspicious more easily, and expect to be treated with respect. This is accomplished by having content authors construct examples of dialog at different levels of difficulty, and using THESPIAN [21] to train PsychSim models of nonplayer character behavior separately for each level of difficulty. Also, the degree of complexity of the language increases steadily as the learner progresses through Mission Game scenes and Arcade Game levels.

2.5. Fish tanks and sandboxes

Gee [4] points out that good games often provide "fish tanks" (stripped down versions of the real game, where gameplay complexity is limited) and "sandboxes" (versions of the game that have similar gameplay to the real game, but where there is less likelihood for things to go wrong). These help users to develop their skills to the point where they can meet the challenges of the full game.

Fish tank and sandbox modes are both provided by the TLTS. An interactive tutorial lets learners practice operating the game controls, and utter their first words of Arabic (/marHaba/ or /as-salaamu 9aleykum/, depending upon the dialect being studied). The Novice mode described above provides sandbox capability. In addition, simplified interactive dialogs with friendly game characters are inserted into the Skill Builder lessons. This enables the learner to practice their conversational skills in a controlled setting.

Finally, sandbox scaffolding is provided in the Mission Game in the form of the virtual aide who can assist if the learner gets stuck. For reasons described above, we avoided having the aide interrupt with tutorial feedback that disrupts gameplay. The aide does not intervene unless the learner repeatedly fails to say something appropriate (this is often a microphone problem that has nothing to do with the learner's actual speech). In this case, or when the learner explicitly requests help, the Pedagogical Agent that drives the animated aide's behavior queries PsychSim for an appropriate user action, and then explains how to perform or say that action in Arabic. PsychSim maintains an agent model of normative user behavior for this purpose, alongside its models of nonplayer behavior.

2.6. Story and character identification

An important aspect of modern serious games is their use of story and character to maintain user interest, and to encourage the user to identify with the game character. Gee [4] has noted that it is not necessary to use virtual reality displays in order to immerse game players in a game. Gamers tend to identify with the protagonist character that they are playing, in a game such as Lara Croft. This is evidenced by the fact that nonplayer characters address either the player's character or the player directly, without seeming contradiction. Identification between player and character is reinforced in the TLTS by the fact that the player speaks on behalf of his character as he plays the game. Feedback from TLTS users suggest that this effect could be enhanced by allowing users to choose their character's uniform, and by adjusting mission and instructional content to match the learner's job, and we plan to provide such customizability in future work.

The TLTS makes extensive use of story structure; the game scenes fit within an overall narrative. This helps maintain learner interest. Also, it is our intention in the TLTS to make it so that actions earlier in the game can have effects on game play later in the game. If for example the player does a good job of developing rapport with characters in the game, those characters are more likely to assist the player later on in the mission. This will help reinforce gameplay to orient the learner toward future game objectives.

2.7. Fun and learning orientation

One of the most important characteristics of a good game, of course, is that it be fun [17]. The fun element helps maintain learner interest and positive attitude, and promotes intrinsic motivation. Evaluations of the TLTS suggest that fun plays an important role in promoting a learning orientation. Consider for example the following quote from a test subject:

> Had I spent more time with the Skill Builder… I probably would have been able to shoot through [the MPE] with relatively little problem. This is kind of fun, the Skill Builder is not that fun. (laughs)

We cannot hope to make the drill and practice exercises of the Skill Builder fun, but if the game component of the learning environment is fun, then learners will engage in the other learning activities in the environment, as resources that help them develop knowledge and skills that are relevant to the game. Squire and Jenkins [22] make a similar observation regarding the serious games that they have developed at MIT.

Thus, the fun element of the games in TLTS sets the stage for serious study and practice in the Skill Builder, provided that learners understand how that study and practice can help them improve their game skills. We can and do apply a wide range of intelligent tutoring and learner modeling techniques in this context. Each Skill Builder lesson includes a variety of different lesson and exercise types. Passive Dialogs show typical dialogs between Arabic-speaking game characters, in a context that is similar to the task context that they are training for. Vocabulary pages introduce words and phrases, and give the learner practice is saying them. A disfluency analyzer analyzes the learner's speech for common pronunciation errors, and then provides coaching on those errors. The feedback is intended to motivate and encourage the learner [10]. The vocabulary pages first show both English translations and Arabic transliterations for the target utterances; these are immediately followed by pages in which the transliterations are removed, in order to make sure that the learner is committing the new vocabulary to memory. Utterance formation exercises require learners to think of an Arabic phrase to say in a particular context, and give them feedback as to whether or not the phrase was appropriate. Active dialogs are similar to passive dialogs, but where the learner plays the role of one of the characters in the conversation. Finally, learners complete a quiz consisting of similar exercise pages, to show that they have mastered the material. They are encouraged to retry these quizzes on subsequent days until they demonstrate full mastery.

One type of AIED processing that we have not found to be of great importance yet is curriculum sequencing. We expect TLTS learners to be motivated to improve their game skills. To assist them in this activity, we provide them with a Skill Map, which shows them what skills they need to master in order to complete each Mission Game scene, and where to find relevant lesson materials in the Skill Builder. We plan to test this new capability in upcoming evaluations, to assess whether this alone is sufficient to provide guidance. If not, we will augment the Skill Map with automated assessments of whether or not they have demonstrated master of each skill, and recommendations of lessons to study or review.

Another role that fun element plays in TLTS, particularly in the Arcade Game, is to provide learners with a pleasant diversion from their study. Learners comment that they like being able to take a break from study in the Skill Builder and play a few levels in the Arcade Game. Yet even when they are taking a break in this fashion, they are still practicing their language use. The opportunity for learners to change pace in this way enables learners to spend many hours per day using TLTS without much boredom or fatigue, something that very few intelligent tutoring systems can claim.

3. Game Development

The TLTS makes use of an existing game engine from Epic Games called the Unreal Engine. A game engine refers to the set of simulation codes that does not directly specify the game's behaviour (game logic) or the contents of the game's environment (level data), but is responsible for visual and acoustic rendering as well as basic interaction such as navigation and object collision. Such engines are increasingly being employed by researchers as affordable and powerful simulation platforms [15]. What makes this technology especially appealing is that games, when purchased off-the-shelf, often include, free of charge, all the authoring tools necessary to create new game logic and level data. Serious games can therefore be crafted from games originally intended for entertainment, avoiding initial game engine development costs.

In TLTS we take a step further by interfacing the Unreal Engine with our own Mission Engine (ME) [8] through the Game Bots interface (http://www.planetunreal.com/gamebots/). The ME and its attached modules handle all of our game logic, including interaction with AI and advanced interfaces such as the speech recognizer. The ME is written in Python, which is a powerful scripting language gaining ground in game development, and reads in data such as descriptions of Skill Builder lessons and game scenes in XML format. This combination of scripting and XML processing enables flexible and rapid development, such as when we added Tactical Iraqi to the existing Levantine Arabic content. Being so heavily data driven, it is essential that we have a good set of data authoring tools. To this end, we have concentrated a good deal of our effort on streamlining the content authoring pipeline and designing tools that are intuitive and effective in the hands of non-programmers. This is important because the game design should not rest on the shoulders of programmers alone, but be a group effort where story writers and artists help enforce proper game design principles.

4. Conclusions

This paper has examined the methods that modern serious games employ to promote engagement and learning, and discusses the role of AIED technology within the context of such games. Serious games can support learning in a wide range of learners, including those who have little initial motivation to study the subject matter. They embody a range of design principles that appear to promote learning, although further evaluative research needs to be done to understand their effects on learning. The serious game context makes the job of the AIED development in many ways easier, since the game design assumes some of the responsibility for promoting learning. AIED development effort can then be focused towards using AI to promote instructive gameplay, managing the level of challenge of the user experience, providing scaffolding selectively where needed, and supporting learners in their efforts to reflect on their play and improve their skills.

Acknowledgments

This project is part of the DARWARS Training Superiority Program of the Defense Advanced Research Projects Agency. The authors wish to acknowledge the contributions of the members of the Tactical Language Team.

References

[1] Burton, R. R., & Brown, J. S. (1982). An investigation of computer coaching for informal learning activities (pp. 79-98). In D. Sleeman and J. S. Brown (Ed.), *Intelligent Tutoring Systems*. New York: Academic Press.
[2] Conati, C. and Maclaren, H. (2004). Evaluating a probabilistic model of student affect. *Proceedings of ITS'04*, 55-66. Berlin: Springer-Verlag.
[3] Csikszentmihalhi, M. (1990). Flow: The psychology of optimal experience. New York: Harper Perennial.
[4] Gee, James Paul. *What Video Games Have to Teach Us about Learning and Literacy.* New York: Palgrave Macmillan, 2003.
[5] Goldstein, I.P. (1982). The Genetic Graph: a Representation for the Evolution of Procedural Knowledge. In D. Sleeman and J.S. Brown, editors, Intelligent Tutoring Systems, pages 51-78. Academic Press, London, 1982.
[6] Hall, L., Woods, S., Sobral D., Paiva, A., Dautenhahn K., Wolke, D. (2004). Designing Empathic Agents: Adults vs. Kids. ITS '04. Berlin: Springer-Verlag.
[7] Hill, R., Douglas, J., Gordon, A., Pighin, F., & van Velsin, M. (2003). Guided conversations about leadership: Mentoring with movies and interactive characters. Proc. of IAAI.
[8] Johnson W.L., Beal, C., Fowles-Winkler, A., Narayanan, S., Papachristou, D., Marsella, S., Vilhjálmsson, H. (2004). Tactical Language Training System: An interim report. ITS '04. Berlin: Springer-Verlag.
[9] Johnson, W.L., Marsella, S., Mote, N., Viljhalmsson, H, Narayanan, S., Choi, S. (2004). Tactical Language Training System: Supporting the rapid acquisition of foreign language and cultural skills. InSTIL/ICALL Symposium, Venice, Italy.
[10] Johnson, W.L., Wu, S., & Nouhi, Y. (2004). Socially intelligent pronunciation feedback for second language learning. ITS '04 Workshop on Social and Emotional Intelligence in Learning Environments.
[11] Johnson, W.L., Beal, C. (2005). Iterative evaluation of an intelligent game for language learning. Proc. of AIED 2005. Amsterdam: IOS Press.
[12] Jones, S. (2003). "Let the Games Begin: Gaming Technology and Entertainment among College Students," Pew Internet & American Life Project, http://www.pewinternet.org/report_display.asp?r=93.
[13] Laird, J. And vanLent, M. (2000). The role of AI in computer game genres. http://ai.eecs.umich.edu/people/laird/papers/book-chapter.htm
[14] Lepper, M. R., & Henderlong, J. (2000). Turning "play" into "work" and "work" into "play": 25 years of research on intrinsic versus extrinsic motivation. In C. Sansone & J. Harackiewicz (Eds.), Intrinsic and extrinsic motivation: The search for optimal motivation and performance (pp. 257-307). San Diego: Academic Press.
[15] Lewis, C., Jacobson, J. (2002) Game Engines for Scientific Research, Communications of the ACM, January 2002.
[16] Norman, D. A. (1990). The design of everyday things. New York: Doubleday.
[17] Prensky, M. (2001). Digital game-based learning. New York: McGraw Hill
[18] Prensky, M. (2002). The Motivation of Gameplay: or, the REAL 21st century learning revolution. On The Horizon, Volume 10 No 1.
[19] Schank, R. & Cleary, C. (1995). *Engines for education*. http://engines4ed.org/hypermedia
[20] Serious Games Initiative. http://www.seriousgames.org.
[21] Si, M. & Marsella, S. (2005). THESPIAN: An architecture for interactive pedagogical drama. Proc. Of AIED 2005. Amsterdam: IOS Press.
[22] Squire, K. & Jenkins, H. (in press). Harnessing the power of games in education. *Insight*.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Taking Control of Redundancy in Scripted Tutorial Dialogue [1]

Pamela W. JORDAN [2], Patricia ALBACETE and Kurt VANLEHN
Learning Research and Development Center, University of Pittsburgh

Abstract. We describe extensions to a finite-state dialogue manager and its author scripting language that enable control over content repetition during tutorial dialogue. The problem of controlling redundancy requires the specification of complex conditions on the discourse history that are beyond the capabilities of the intended users of the scripting language. We show that the problem can be addressed by adding semantic labelling and the marking of optional steps and difficulty levels to the scripting language and heuristic algorithms to the dialogue manager.

Keywords. Dialogue Management, Authoring tools

1. Introduction

One of the many challenges in building intelligent tutoring systems that interact with students via natural language dialogue is selecting a dialogue management approach for which course content can be easily authored by non-technical users while still maximizing adaptability to the context. Our initial approach to dialogue management in the WHY-ATLAS tutoring system [13] focused on simplifying the authoring task and can be loosely categorized as a finite state model. Finite state models are appropriate for dialogues in which the task to be discussed is well-structured and the dialogue is to be system-led [8] as was the case for WHY-ATLAS. This initial approach proved successful with regard to ease of authoring [4,10] but ultimately proved not to be sufficiently sensitive to the discourse context.

Instructors use a scripting language [4] to author the content and form of the finite state network that represents the space of dialogue moves the tutor can make given the student's response to a previous move. The scripting language allows the author to specify a multi-step hierarchically-organized recipe (a type of plan structure defined in AI planning) for covering a topic or part of the problem solving for which the student needs help. Recipes are higher level goals that are defined as a sequence of any combination of primitives and recipes [16]. In WHY-ATLAS a step which is a primitive is a state in the finite state network. The author also specifies classes of answers to any tutor questions asked in a primitive and appropriate state transitions for each class of answer. Thus each answer is an arc between two states in the network. By default, if no transition is specified for a step then the transition is to the next step in the recipe. Authors can label

[1] This work was funded by NSF grant 0325054 and ONR grant N00014-00-1-0600.

[2] Correspondence to: 3939 O'Hara Street, Pittsburgh PA, 15260, USA; E-mail: pjordan@pitt.edu.

primitives with goals but the scripting language does not distinguish state transition information from goal information at the primitive level. This means that the goal labels for primitives must be unique if the originally scripted line of reasoning is to be recovered on-demand from the network.

When an author scripts dialogues to support tutoring for multiple problems that the student is to solve, the author should not pre-suppose what will have been seen by the student or how well the student responded to previous questions. Such reactions need to be encoded as conditions on the discourse context. However, adding conditions to the scripting language moves the authoring task closer to a programming task and potentially makes it too difficult for many instructors.

While we initially chose to ignore the need to specify more complex conditions on the context in order to make the task one that any instructor is likely to be able to do, the trade-off is redundancy in the material discussed with a student. Since students work on multiple problems during tutoring and all these problems share some subset of domain concepts with at least one other problem, a student might see similar content many different times. Although redundancy can be beneficial, if used inappropriately it can be detrimental to attention and to the quality of the solutions produced during problem solving [14,6].

During reviews of the WHY-ATLAS transcripts[1], we found that when the system repeats content (whether in the same problem or across problems), students will often still answer but will also additionally append insults or displays of annoyance ("up, you idiot", "same, like I said"), or expressions of confusion ("i don't know what u want me to say."). Or they may suspect that they are being misunderstood and try to solve the problem by doing such things as oversimplifying responses ("lightweight car massive truck patch frictionless ice head-on collision vehicle impact force greater change motion"). At other times they simply stop answering ("I don't know" or null response). The loss of motivation and the loss of credibility of a tutor are expected to have some detrimental effect on learning.

Our solution for controlling redundancy is to share the task of specifying conditioning on context between the author and the dialogue management software by making the author's added task one of labelling rather than of programming. Authors are asked to optionally label dialogue moves with similar content with a consistent labelling scheme of their own choosing and mark the difficulty level of a move relative to those with similar labelling. Given this additional information and heuristic algorithms, the dialogue manager has the additional information it needs to more wisely use redundancy. It can now check the dialogue history for previous uses of a label and find out how often the content has been presented and how well the student responded in each of those cases. It allows the dialogue manager to either skip moves or to select moves that are more or less challenging based on the student's previous performance on that same labelled move. This addition is similar to what is suggested in contingent tutoring [15].

In this paper, we focus on the changes we have made to the scripting language and to the dialogue manager. First we review the WHY-ATLAS system and the old scripting language and dialogue manager. Next we describe the extensions to the scripting language and how the dialogue manager uses this additional information to provide additional conditioning on the context. During the discussion we show two examples of op-

[1] We reviewed 110 system-student dialogue sessions in which one session covers one physics problem. All quoted examples are verbatim from these transcripts.

tionally enhanced scripts and how they adapt to the context. We conclude with a preliminary evaluation of instructors' ability to used the extended scripting language and our plans for evaluating the effectiveness of the resulting dialogues.

2. The WHY-ATLAS System

Question: Suppose you are running in a straight line at constant speed. You throw a pumpkin straight up. Where will it land? Explain.

Explanation: While I am running, both the pumpkin and I are moving with the same speed. Once I throw the pumpkin up, it no longer has anything to thrust it forward in the either the horizontal or vertical direction. Therefore, it will fall to the ground behind me.

Figure 1. The statement of the problem and a verbatim student explanation.

The WHY-ATLAS system covers 5 qualitative physics problems on introductory mechanics. When the system presents one of these problems to a student, it asks that she type an answer and explanation and informs her it will analyze her final response and discuss it with her. One of the problems WHY-ATLAS covers is shown in Figure 1 and the student response shown is a first response from our corpus of students' problem-solving sessions. The student response in this case is an instance of the often-observed *impetus* misconception: If there is no force on a moving object, it slows down. In a majority of student responses, the only flaw is that the response is incomplete. Details about how the essay is analyzed are addressed in [7,5,9] and are beyond the scope of this paper.

2.1. Dialogue Management

Given the results of the essay analysis, which is a list of topic labels, the WHY-ATLAS dialogue subsystem leads a discussion about those topics. It uses a robust parsing approach (CARMEL [11]) to understand the student's input and match it to the expected inputs, and a reactive planner (APE [2]) to manage the dialogue where choosing the next dialogue move depends upon the student's answer.

There are 3 types of dialogue recipes that were scripted for WHY-ATLAS; 1) a walkthrough of the entire problem solution, 2) short elicitations of particular pieces of knowledge or 3) remediations. Walkthrough recipes are selected when the student is unable to provide much in response to the question or when the system understands little of what the student wrote. Short elicitations are selected if the student's response is partially complete in order to encourage the student to fill in missing pieces of the explanation. Remediations are selected if errors or misconceptions are detected in the student's response to the question. During the course of the top-level recipe type selected, pushes to recipes for subdialogues that are of the same three types (i.e. walkthrough, elicitation or remediation) are possible but typically are limited to remediations.

After the discussion based on the top-level recipe is complete (may have pushed to and popped from many recipes for subdialogues during the course of the main recipe), the system will either address an additional fault in the essay or ask that the student revise her explanation before moving on to any other flaws already identified. The cycle of explanation revision and follow-up discussion continues until no flaws remain in the student's most recent essay.

2.2. Dialogue Scripts

Dialogues are represented as finite state networks with a stack (i.e. a pushdown automaton). States correspond to primitives that produce tutor utterances, arcs correspond to correct student responses or cases in which no response is expected, and pushes to vague or incorrect student responses. Pushes call a subdialogue and pops return from one.

The scripting language defines primitive actions and recipes. A primitive is defined to be a tutoring goal that is a leaf node in a plan tree and an associated natural language string that realizes that primitive tutoring goal. A primitive may encode a tutor explanation or a question for eliciting a particular piece of information or both.

Recall that recipes are higher level goals that are defined as a sequence of any combination of primitives and recipes [16]. This representational approach is widely used in computational linguistics since problem-solving dialogues and text are believed to be hierarchically structured and to reflect the problem-solving structure of the task being discussed [3]. Tutorial intentions or goals should be associated with both recipes and primitives. In this way, the author may encode alternative ways of achieving the same tutorial intention.

For each primitive tutoring goal, the scripting language also includes information on what to expect from the student so that information on how to respond appropriately to the student can also be included in the script. Possible student responses are categorized as expected correct answers, vague answers and a set of expected typical wrong answers. For completeness, the author is expected to always include a class for unrecognized responses as well. Every vague and wrong answer and the default class have associated with them a list of tutorial goals that must be achieved by the dialogue manager in order to respond appropriately to that answer class.

3. Controlling Redundancy

What is redundant depends on the student's history so the goal is to adequately track content across all tutoring sessions with a student. We have added three types of optional information to the scripting language that will help with tracking content and controlling redundancies: 1) semantic labels 2) optional steps within a multi-step recipe 3) difficulty levels for recipes and primitives. We will discuss each in more detail below.

3.1. Semantic Labels and Optional Steps

The original scripting language denigrates the goal labels for primitives with respect to their planning origins by collapsing goal labels and arc pointers. This was done mostly because authors had difficulty associating a goal with every step and found it easier to think of these labels as pointers. But an arc pointer is limited to a specific state while goals are meant to be relevant to multiple states. Thus not only is the power of multiple ways of achieving a goal lost at the primitive level so is the knowledge that primitives from different recipes may cover similar content.

Because the dialogue manager does not have any information on the meaning of the content encoded in the network, it cannot detect repetitions with sufficient reliability to reduce repetition or possibly even push the students with increasingly more challenging questions. So while the dialogue manager does track the dialogue history by recording 1) what has been contributed to the dialogue by both the student and tutor, and 2) the language interpreter's classification of student responses to tutor questions, it does not have access to the meaning of the tutor's turn. So context conditioning is strictly limited to the previous student response, and the dialogue manager can not skip steps that were made obsolete by its own previous actions or earlier student responses.

To solve this problem, we added semantic labels for primitives and recipes so that all primitives and recipes with similar content are recognizable to the dialogue manager and we added markers for optional steps that can be skipped given the proper context. The semantic labels used are up to the author. The author can make the label meaningful to him or not (e.g. elicit-force vs sem1) but it has no actual meaning to the system. The system only looks for exact label matches between a turn that is about to be delivered and one or more previous turns.

We cannot always skip redundant material because redundancy has a beneficial role to play in task-oriented dialogue. The more relevant roles for tutoring are that it either brings a piece of knowledge into focus so that inferences are easier to make, or emphasizes a piece of knowledge. We also know that for learning, repetition of a piece of knowledge that a student is having difficulty grasping is sometimes helpful. Given these roles, the location of the redundancy with respect to time and how the student previously performed are considered.

As an example, the following script includes semantic labels (i.e. :sem <label>) and optional steps (i.e. :step*). Here we will assume the remediation recipes each use the same labels for semantics as for goal names.

```
(goal detailed-analyze-forces
      :sem detailed-analyze-forces
      (:step
       "Try to name all the forces acting on the pumpkin after
        it is thrown."
        :answers
        (("gravity")("air resistance" remind-negligible)
         ("$anything else$" help-id-forces)))
      (:step*
       :sem help-id-forces
       "Are there any other forces on the pumpkin?"
       :answers
       (("no")("$anything else$" help-id-forces)))
      (:step*
       :sem remind-negligible
       "Why is there no force on the pumpkin due to air?"
       :answers
       (("negligible")("$anything else$")))
      (:step "So gravity is the only force on the pumpkin")
      ....)
```

Here the second and third steps are marked as optional. There are two ways in which an optional step can be skipped. The first is if a semantic label is in the immediate discourse history. In the above, the semantic label *help-id-forces* would be in the immediate discourse history if the student's answer in the previous step was not recognized (i.e. categorized as answer class "$anything else$") and a push was made to the remediation recipe for that class, *help-id-forces*. The same is true for *remind-negligible*. The second way of skipping is if the semantic label is in the non-immediate history and the student did well with it when last encountered.

3.2. Levels of Difficulty

While we know that repetition of difficult material can be beneficial, it should be gradually adjusted over time so that the student is providing the knowledge with decreasing assistance from the tutor. In addition the tutor could try to help the student achieve a deeper understanding. To address these possibilities, we added the specification of difficulty levels at the primitive and recipe levels to work in conjunction with semantic labels and optional steps. To encode difficulty levels, we use speech act labels to distinguish primitives with the same semantic labels and intent levels for recipes with the same semantic labels.

A speech-act [12] is a type of intention behind an utterance. The two most frequently discussed speech-acts in the literature are inform and request [1]. Inform is frequently realized as a declarative sentence while a request is frequently realized as a question. For tutoring, we further sub-divide the request speech-act by question-type and use the labels "Whyq" for why questions, "howq" for how questions, "ynq" for yes/no questions, and "whq" for all other questions (i.e. when, where, what). An example in which different questions types are used that involve the same concept is: "Why is the pumpkin's acceleration downward?" vs. "Does the pumpkin have a downward acceleration?" vs. "What is the direction of the pumpkin's acceleration?" vs. "The pumpkin accelerates downward."

If primitives with the same semantic label are defined using different speech-act/question-types, the semantic label is already in the discourse history and the student was successful with the previous form, then a "harder" speech-act/question-type is selected (if it is available). The first speech-act/question-type defined for a step is the default one if the semantic label is not yet in the student's history. Otherwise the question-types are organized by difficulty as follows (howq whyq whq ynq) from hardest to answer to easiest. So if the student always has trouble with the whyq and higher for a particular semantic label then the question-type that is selected (if available) is whq or lower. If the student got the previous question-type right for a semantic label then the selection heuristic will try the next hardest type available. Note that if the student gets it wrong then an easier type will be tried the next time and then a harder one the next if she is able to answer it. If the hardest question-type specified was previously tried and the student got the question right then the student will get an inform if one is available (the assumption is that she must already know this bit of knowledge now).

When multiple recipes have the same semantic label, an intent label indicates the difficulty level of the recipe. In this case, higher numbers indicate increased difficulty and 0 is reserved for a recipe that simply informs. For example, below are two recipes for goal G with the same semantic label *a*, two recipe intent levels (i.e. :intent <level>) and two speech-act/question-types for one step in the second recipe (i.e. :sa <speech-act/question-type>).

```
(goal G
  :sem a
  :intent 0
  "After the object is released, the only force acting on it is
   gravity. This force is called weight and is always present
   when an object is in a gravitational field.")
```

```
(goal G
  :sem a
  :intent 1
   (:step
   "What force is responsible for an object's weight?"
    :answers
    (("gravity")("$anything else$" forces-in-a-freefall-inform))))
   (:step
    :sa inform
     "The force of gravity is always present when an object is in
      a gravitational field such as the one produced by earth."
    :sa whq
     "When is gravity present?"
     :answers
     (("in gravitational field")
     ("$anything else$" gravity-near-earth-inform)))
   ...)
```

The first time the recipe for goal G is initiated, label *a* is not in the discourse history. Thus the student gets G with intent level 1 and an inform for the second step which is the default since it is listed first. The second time that she needs G, she will get the intent level 1 version and the whq for the second step. If she needs G a third time, assuming she got all of the steps in the last version of G right, she will get intent level 0 with the assumption being that the content denoted by label *a* is now known and just needs to be brought into focus.

4. Preliminary Evaluation and Conclusion

Two instructors who had previously used the original scripting language were asked to author new material for an upcoming experiment to compare tutoring systems and were asked to try to use the new options to reduce redundancy. Together they authored approximately 350 new recipes and added optional difficulty levels to 11% of these recipes. They also authored 645 new primitives and added semantic labels to 20% of these new recipes and primitives. Finally they marked 4% of the new primitives as optional steps. No optional question types were used. The instructors considered optional question types a low priority and ran out of time before having a chance to try using them.

The enhanced scripts are currently in use in the WHY-ATLAS system and when the current experiment is completed we will analyze these new dialogue transcripts to see if the reactions to better controlled redundancy are neutral as opposed to negative. In future experiments we will compare the learning gains of the enhanced dialogues to unenhanced ones.

We presented an enhanced dialogue manager and scripting language that is sensitive to scripted redundancy in a way that is theoretically beneficial to tutoring. We presented examples of enhanced scripts and discussed how they control redundancy. A preliminary evaluation of the extended scripting language showed that instructors were able to make immediate use of all but one new option. This suggests that we have met our goal of keeping the scripting task from becoming a programming task so that it is still doable by most instructors.

References

[1] Philip Cohen and Raymond Perrault. Elements of a Plan-Based Theory of Speech-Acts. *Cognitive Science*, 3:177–212, 1979.

[2] Reva Freedman. Plan-based dialogue management in a physics tutor. In *Proceedings of the 6th Applied Natural Language Processing Conference*, 2000.

[3] Barbara Grosz and Candace Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12:175–204, 1986.

[4] Pamela Jordan, Carolyn Rosé, and Kurt VanLehn. Tools for authoring tutorial dialogue knowledge. In *Proceedings of AI in Education 2001 Conference*, 2001.

[5] Pamela W. Jordan, Maxim Makatchev, and Kurt VanLehn. Combining competing language understanding approaches in an intelligent tutoring system. In *Proceedings of the Intelligent Tutoring Systems Conference*, 2004.

[6] Pamela W. Jordan and Marilyn A. Walker. Deciding to remind during collaborative problem solving: Empirical evidence for agent strategies. In *Proceedings of AAAI-96*. AAAI Press, 1996.

[7] Maxim Makatchev, Pamela Jordan, and Kurt VanLehn. Abductive theorem proving for analyzing student explanations and guiding feedback in intelligent tutoring systems. *Journal of Automated Reasoning: Special Issue on Automated Reasoning and Theorem Proving in Education*, 32(3):187–226, 2004.

[8] Michael McTear. Spoken dialogue technology: enabling the conversational user interface. *ACM Comput. Surv.*, 34(1):90–169, 2002.

[9] Uma Pappuswamy, Dumisizwe Bhembe, Pamela W. Jordan, and Kurt VanLehn. A multi-tier NL-knowledge clustering for classifying students' essays. In *Proceedings of 18th International FLAIRS Conference*, 2005.

[10] Carolyn Rosé, Pamela Jordan, Michael Ringenberg, Stephanie Siler, Kurt VanLehn, and Anders Weinstein. Interactive conceptual tutoring in Atlas-Andes. In *Proceedings of AI in Education 2001 Conference*, 2001.

[11] Carolyn P. Rosé. A framework for robust semantic interpretation. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 311–318, 2000.

[12] John R. Searle. What Is a Speech Act. In Max Black, editor, *Philosophy in America*, pages 615–628. Cornell University Press, Ithaca, New York, 1965. Reprinted in *Pragmatics. A Reader*, Steven Davis editor, Oxford University Press, 1991.

[13] Kurt VanLehn, Pamela Jordan, Carolyn Rosé, Dumisizwe Bhembe, Michael Böttner, Andy Gaydos, Maxim Makatchev, Umarani Pappuswamy, Michael Ringenberg, Antonio Roque, Stephanie Siler, and Ramesh Srivastava. The architecture of Why2-Atlas: A coach for qualitative physics essay writing. In *Proceedings of Intelligent Tutoring Systems Conference*, volume 2363 of *LNCS*, pages 158–167. Springer, 2002.

[14] Marilyn A. Walker. *Informational Redundancy and Resource Bounds in Dialogue*. PhD thesis, University of Pennsylvania, December 1993.

[15] David Wood. Scaffolding, contingent tutoring and computer-supported learning. *International Journal of Artificial Intelligence in Education*, 12:280–292, 2001.

[16] R. Michael Young, Martha E. Pollack, and Johanna D. Moore. Decomposition and causality in partial order planning. In *Second International Conference on Artificial Intelligence and Planning Systems*, 1994.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Ontology of Learning Object Content Structure

Jelena JOVANOVIĆ, Dragan GAŠEVIĆ
FON – School of Business Administration, University of Belgrade
Jove Ilica 154, 11000 Belgrade, Serbia & Montenegro

Katrien VERBERT, Erik DUVAL
Dept. Computerwetenschappen, Katholieke Universiteit Leuven
Celestijnenlaan 200A, B-3001 Leuven, Belgium

Abstract. This paper proposes an ontology that enables a formal definition of Learning Object (LO) content structure. The ontology extends the Abstract Learning Object Content Model (ALOCoM) with concepts from information architectures. It defines a number of concepts that represent different types of content units and it specifies their structure. Formalising structural aspects of LOs, the ontology facilitates re-purposing of LOs at different levels of content granularity, i.e. LOs in their entirety and their components. Furthermore, being a generic LO content model, the ontology serves as an integration point of heterogeneous LO content models.

Introduction

There is an increasing interest in the learning technology community for repurposing learning objects (LOs) [1]. Presently, authors of learning materials employ a cut & paste approach when composing new LOs out of components of existing ones. Nonetheless, such an approach is non-scalable in terms of maintenance, since each time you copy a content unit, you create a new place that needs to be maintained [2]. Additionally, the process tends to be error-prone, and due to its inherent monotony, easily becomes both bothering and time consuming. The authors are in a much better position if access to the components of LOs and their composition into meaningful units is made, at least partially, automatic. A possible solution employs a more reusability prone format of LOs that makes their structure explicit and thus enables reusability of LO components as well. This can be accomplished through provision of a flexible model of LO content structure. An explicit content structure allows the disaggregation of a LO into its constituent components. Those components, enriched with fine-grained descriptions (metadata), increase the findability of relevant content units.

Ontologies and Semantic Web technologies can be a solid basis for solving the aforementioned problem, as an ontology gives a formal specification of the shared conceptualization of a certain domain. For the domain of e-learning, we found a classification of ontologies suggested in [3] relevant. The classification differentiates between: a) *content* (*domain*) ontologies describing the subject domain of a content unit, b) *context* (*didactic*) ontologies formally specifying the educational/pedagogical role of a content unit, c) *structure* ontologies providing a shared conceptualization of how content units can be assembled together to form a coherent learning whole.

High level of LO re-purposing can be achieved if learning materials are broken down into small content units that can be easily handled. Accordingly, concepts from the structure ontology are especially useful. If we have LO repositories with learning content disaggregated to content units of the lowest level of granularity (e.g. a single image, text fragment or audio/video clip) and presented in a structure ontology-aware format, we will be able to make the process of composing new learning materials out of components of existing LOs (partially) automatic. Furthermore, this structure related information would also be of great importance to a dynamic assembly engine of an Adaptive Learning System when combining content units into a meaningful and well structured learner tailored presentation.

In this paper, we present an ontology that we propose for the formal specification of LO content structure. The ontology extends the Abstract Learning Object Content Model (ALOCoM) that defines a framework for LOs and their components [4], with concepts from the Darwin Information Typing Architecture (DITA) – an XML-based architecture for authoring, producing, and delivering technical information that is easy to reuse [2].

The paper is organized as follows: in the next section we give a concise overview of the conceptual origins of the ALOCoM ontology and we briefly describe the ontology architecture. In the second section we explain the ontology implementation in detail. Section 3 explains the enabling role that the ontology has in achieving interoperability among different content models and Section 4 concludes the paper.

1. Conceptual Solution

This section explains the conceptual origins of the ontology, thus enabling easier comprehension of the ontology architecture and design.

The Ontology Origins

As we stated in the introduction, the proposed ontology is a generic content model that defines a framework for LOs and their components [4]. As Figure 1 suggests, the model differentiates between Content Fragments (CF), Content Objects (CO), and Learning Objects (LO).

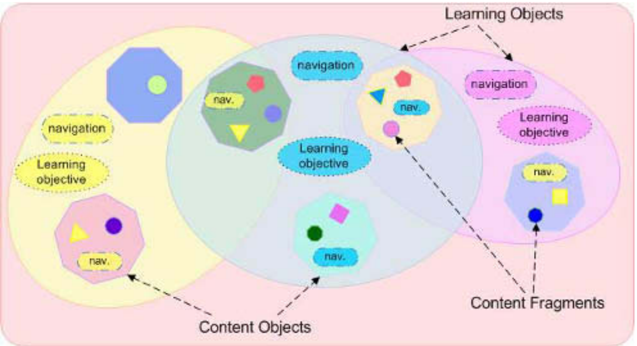


Figure 1. A sketch of Abstract Learning Object Content Model

CFs are content units in their most basic form, like text, audio and video. Basically, CFs are raw digital resources. They can be further specialized into discrete (graphic, text, image) and continuous (audio, video, simulation and animation) elements. COs aggregate CFs and add navigation. Navigation elements enable proper structuring of CFs within a CO. Besides CFs, a CO can include other COs as well. At the next aggregation level, a LO is defined as a collection of COs with an associated learning objective.

Further, we defined content types for each of these components. We introduced CF types such as image, text, audio and video. For defining CO types, we investigated existing Information Architectures, like the Information Block Architecture [5] developed by Dr. Horn and the IBM Darwin Information Typing Architecture [6]. These architectures define information types (e.g. concept, principle, task) and their building blocks (e.g. example, definition, analogy). As a starting point, we defined the CO types and their structure using DITA concepts, since DITA is a recent architecture with rich documentation and online support [6]. Besides CF and CO types, the ontology identifies LO types such as a Lesson, a Report, a Course and a Test. Finally, the ontology defines the relationships between the LO components. For now, aggregational and navigational relations are specified.

1.2 The Ontology Organization

An important feature of the DITA architecture is the extensibility of the core information types aimed to meet specific needs of an author/community. Since our objective is to have a content structure ontology that supports different kinds of LOs, and that is easily extensible to include new LO types, we decided to make use of DITA's inherent extensibility in the ontology we were developing. Therefore, we organized the ALOCoM ontology as an extensible infrastructure consisting of: the core part (ALOCoMCore) with concepts common for all LO types and an unlimited number of extensions, each extension supporting one specific LO type. Figure 2 illustrates this hierarchical architecture. The main benefits of the proposed, extensible, ontology architecture is to avoid large and clumsy vocabularies: ontology extensions can meet specific requirements of each application domain. In other words, exclusively the ontology extension defined for a specific LO type that the application works with, should be included to avoid unnecessary information burden.

Additionally, the core part of the ALOCoM ontology is an integration point of different LO content models (SCORM, CISCO, Learnativity, etc.). Therefore, we defined extensions of the core ontology that serve as mappings between ALOCoM and other LO content models. This topic is further extended in the section 3.

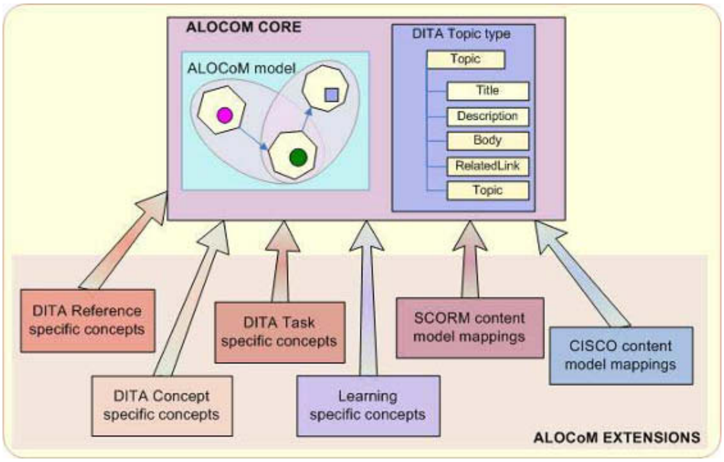

ALOCOM CORE
ALOCoM model
DITA Topic type
Topic
Title
Description
Body
RelatedLink
Topic
DITA Reference specific concepts
DITA Task specific concepts
SCORM content model mappings
CISCO content model mappings
DITA Concept specific concepts
Learning specific concepts
ALOCoM EXTENSIONS

Figure 2. A vision of hierarchical structure of the ALOCoM ontology

2. The Ontology Implementation

We used the Web Ontology Language (OWL) – the W3C recommendation [7] – to develop the ALOCoM ontology and exploited advantages of OWL specific features for ontology development. These features can be summarized as follows:

- Solid modularization mechanism that enables the definition of easily extensible ontologies.
- Support for definitions of concept hierarchies, so that reasoners can recognize the presence of the inheritance (is-a) relationship between two concepts.
- Advanced ways for describing properties like: the range of a property defined as a union of two or more other classes, definition of cardinality restriction, etc.
- Ability to define synonyms, so we can make equivalences (or mappings) between the concepts of two (or more) vocabularies covering the same domain. For example, we can define mappings between ALOCoM and SCORM terminology – e.g. an ALOCoM CF is equivalent to a SCORM Asset.

To implement the ontology, we used the Protégé ontology development tool (http://protege.stanford.edu), since it has support for development, storage and editing of ontologies in OWL format.

In the following subsections we present the ontology in detail. First, we explain the design of the core part of the ontology and then focus on the ontology extensions.

2.1 The Core Ontology

The first step in building the core part of the ontology was to define classes for representing CFs, COs, and LOs in general. Subsequently, we added a number of classes corresponding to the specific types of a LO components (i.e. COs and CFs).

As we stated in section 1.1, the ALOCoM ontology defines a number of CF types divided into two main categories of continuous and discrete CFs. Accordingly, we extended the *ContentFragment* class of the ontology with *ContinuousCF* and *DiscreteCF* classes, respectively representing these two main CF types. The *DiscreteCF* is further specialized into *Text*, *Image* and *Graphic* classes, while the *ContinuousCF* is further extended with *Audio*, *Video*, *Animation* and *Simulation* classes.

Further, we extended the *ContentObject* class of the core ontology with a number of classes representing different kinds of COs that can be part of almost any type of LO. We based those classes on elements of the DITA information architecture. One ontology class is introduced for each DITA element that we found appropriate for describing content units typical for the learning domain. Accordingly, many of the DITA building blocks, such as *section*, *paragraph*, *list* etc., are included in the core ontology as either direct or indirect subclasses of the *ContentObject* class. We did not include those DITA elements that are presentation-related, such as the *searchtitle* element that is used when transforming a DITA element to XHTML to create a title element at the top of the resulting HTML file [6].

One should note that, even though the ALOCoM ontology is based on the DITA model, some of the ontology concepts are not identical in meaning to the corresponding DITA elements. The primary reason for this lies in the obvious discrepancy of the intended application domains of DITA and ALOCoM: while DITA is devised exclusively for the technical domain, the ALOCoM ontology is intended to be used in a variety of learning domains. Therefore, we need to make the structure of certain DITA elements more general, so that they can be applicable not just for structuring of technical information, but also for structuring of content in any other learning domain (e.g. mathematics, arts, etc). Additionally, the structure of certain DITA elements is overwhelmed with presentation-related components (e.g. *table*, *link*, *definitionlist*). Being interested in content structure released from presentation details, we created ontology classes corresponding to a simplified version of such DITA elements (e.g. Link, Definition), leaving out all of their presentation-oriented components. Generally speaking, DITA served us as a good starting and reference point to get an overview of the concepts potentially relevant for an explicit specification of LO structure.

The *LearningObject* class is introduced to represent the LO content type. Descendents of this class are defined in the ontology extensions. Each extension typically covers one specific LO type.

Finally, the core part of the ALOCoM ontology defines several types of properties. From the perspective of content structuring, the following four are the most important: *hasPart*, *isPartOf*, and *ordering*. The definition of these properties is graphically represented in Figure 3, using the Ontology UML Profile – OUP presented in [8].

The *hasPart* and its inverse *isPartOf* properties allow us to express aggregational relationships between content units. The domain of the *hasPart* property is defined as the union of COs and LOs, since CFs represent elementary content units that cannot be formed of smaller meaningful content units. The range of this property is defined as the union of CFs, COs and LOs. We exploited the mechanism of restrictions to constrain the range of this property for almost each type of both COs and LOs. For example, in the case of the *List* CO type, the range of this property is restricted to encompass only instances of the *ListItem* type, or in the case of the *Table* CO type, the range of the same property is restricted to the union of *TableRow*, *TableData* and *Title* classes. Similar restrictions are defined for the *isPartOf* property. In the left part of Figure 4, we used OUP to depict restrictions imposed on the range of the *isPartOf* property in the context of the *ListItem* concept. As the figure shows, the range of the property is limited solely to the instances of the *List* class. The right part of the same figure presents the diagram in the OWL XML binding.

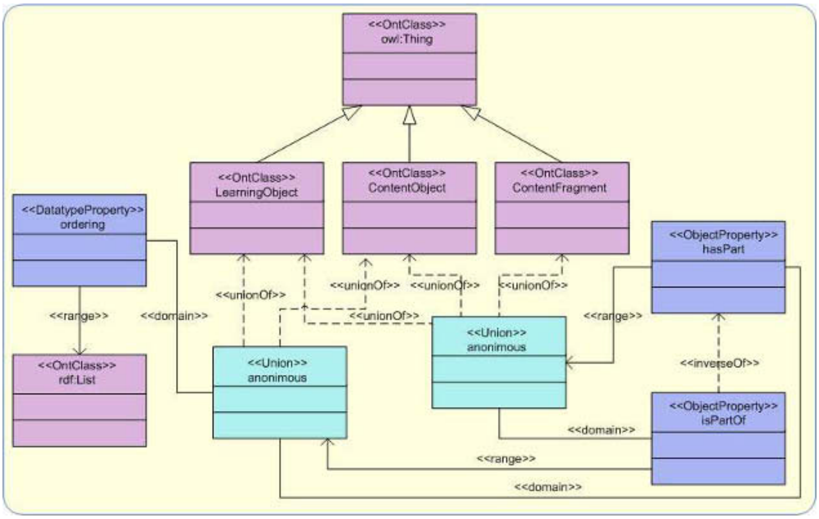
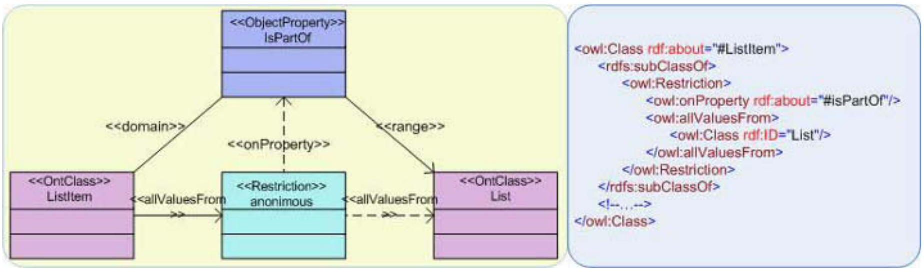


Figure 3. A scatch of major properties of the ALOCoM ontology in OUP

The *ordering* property allows us to express sequencing of components aggregated in a composite content unit (e.g. sequencing of CFs inside a CO). The domain of this property is a union of COs and LOs. CFs are not included in the domain, since CFs are elementary content units that cannot be further dissagregated. The range of this property is an rdf:list consisting of identifiers of components belonging to the composite content unit. The order of these identifiers in the rdf:List defines the order of components in a composite content unit (i.e. CO or LO). The elements of such an rdf:List must be identifiers of the resources that form the range of the *hasPart* property of the composite content unit. A composite content unit can have an arbitrary number of ordering properties, each one defining a specific learning path.



```
<owl:Class rdf:about="#ListItem">
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:about="#isPartOf"/>
            <owl:allValuesFrom>
                <owl:Class rdf:ID="List"/>
            </owl:allValuesFrom>
        </owl:Restriction>
    </rdfs:subClassOf>
    <!--. . .-->
</owl:Class>
```

Figure 4. Restriction on the range of the *isPartOf* property of the *ListItem* class

2.2 The Ontology Extensions

As it was previously stated, the ALOCoM ontology is organized as an extensible architecture. Each extension of the core part of the ontology introduces a set of classes representing content units specific for a certain content type. Up till now, we defined three extensions, namely ALOCOMConcept, ALOCoMTask and ALOCoMReference, each one corresponding to a DITA core information type (*concept*, *task* and *reference* respectively). Due to the space limit, we shall briefly describe just one of those extensions, ALOCoMTask. Within this extension, we introduce classes corresponding to the content units specific for the DITA *task* information type. *Task* generally provides step-by-step instructions explaining how to perform certain task, i.e. what to do and in which order [6]. In Figure 5 the ontology classes introduced in this extension are presented in violet (*Task*, *TaskContext*, *TaskPrereq*, *TaskPostReq*, *TaskBody*, *Info*, *Command*, *Choice*, *Step*, *Result*), while concepts from the core ontology are in dark blue (*owl:Thing*, *LO*, *CO*, *Topic*, *Body*, *CF*).

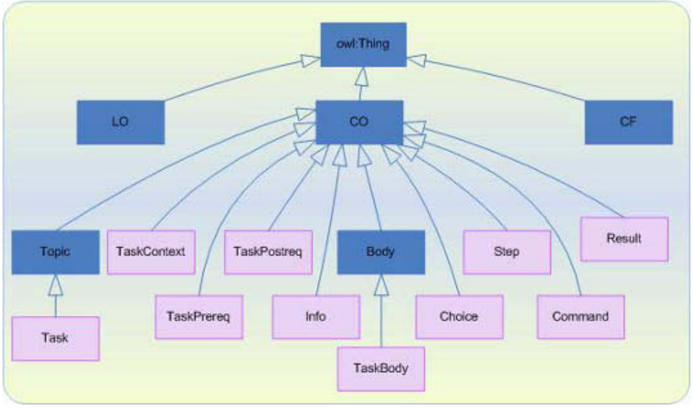


Figure 5. ALOCoMTask ontology extension

Since our intention is to enable content structuring in the learning domain, we are naturally interested in enriching the ontology with additional classes representing content units common to learning situations. Therefore, we are currently developing an extension, named ALOCoMLearning (Figure 6). We introduced, among others, a question, answer and exercise building block, since these content units are typical for learning. DITA does not provide these building blocks as the intent of DITA is primarily technical documentation. Furthermore, classes such as *Lesson*, *Test* and *Course* are defined as new types of LOs.

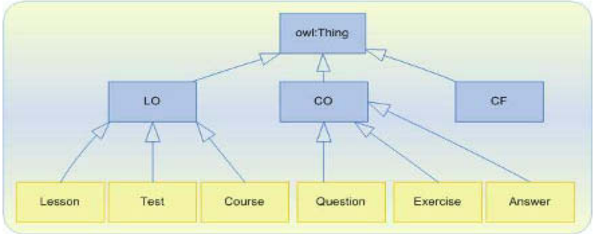


Figure 6. ALOCoM ontology extension with learning-specific classes

3. Ontology-based content model mappings

The semantic heterogeneity of LO content models (e.g. a SCORM Asset is equivalent to a CISCO Content Item) prevents us to automate the process of assembling a new LO out of content units defined in compliance with different content models. Accordingly, there is a need for a generic LO content model that would enable reuse and repurposing of content units developed according to one content model in the context of another one. The ALOCoM ontology, being built on such a generic model, has a potential to serve as a mediator, enabling communications between disparate LO content models.

We base our approach on a method proposed in [9] for integrating data using ontologies. The method has three main stages: building a shared vocabulary, building local ontologies and defining mappings. We have developed the ALOCoM ontology that has the role of a shared vocabulary, as well as one (local) ontology for each investigated LO content model (SCORM, CISCO, Learnativity, NCOM, NETg) and we defined mappings between the global and local ontologies. Table 1 gives a rough overview of those mappings. The next step is to implement those mappings so that resoners can use them to perform automatic translations between different content models. Since both global and local ontologies are written in OWL, we used the *owl:equivalentClass* property to express semantic equivalences between concepts from the global and local vocabularies. However, mappings implemented in such a way are sufficient for some simple reasonings, but in some situations we would need a more expressive mechanism [10]. Therefore, we are considering using RuleML (http://www.dfki.uni-kl.de/ruleml/) or the Semantic Web Rule Language – SWRL [11], as declarative languages for expressing rules, in this case transformation rules. An alternative would be to use a Java-based framework for the Semantic Web (e.g. Jena, http://jena.sourceforge.net/) that provides a Java API for working with ontologies.

Table 1. An overview of mappings between analyzed LO Content Models and ALOCoM

ALOCoM	Content Fragment	Content Object	Learning Object			
Learnativity	Raw Media Element	Information Object	Application Specific Object			
			Aggregate Assembly			
			Collection			
SCORM	Asset	Sharable Content Object	Content Aggregation			
CISCO	–	Content Item	Reusable Information Object		Reusable Learning Object	
		Practice Item				
		Assessment Item				
NETg	–	–	Topic	Unit	Lesson	Course

4. Conclusions

In this paper, we presented the ALOCoM ontology that we developed to provide a more explicit specification of the structure of learning content units. With such an ontology we are able not only to reuse complete learning units, but also to reuse their components. To build the ontology we used some concepts form the DITA architecture, while we adapted some of them to better support the e-learning domain. The ALOCoM ontology is organized as an extensible architecture comprising one core part with the concepts common for all LO types and an unlimited number of extensions for each supported LO type. Apart from defining the common concepts in the ontology core, we defined semantic equivalencies between the ALOCoM ontology and several well-known content models (e.g. SCORM, CISCO, etc.).

We regard the ontology as a promising starting point for our further research towards achieving automated mappings between the most important content models as well as different LO types. We are currently setting up an ALOCoM ontology based LO repository and framework [12] that we are going to use for performing experiments on the ontology. Our goal is to evaluate to what extent the ontology can be used as a mediator for bridging different content models. We are also planning to extend the ontology by using some of Semantic Web rule languages (e.g. RuleML) in order to have more precise mappings between ALOCoM and other content models.

References

[1] Duval, E. and Hodgins, W., “A LOM research agenda”, *In Proceedings of the 12th International World Wide Web Conference*, Budapest, Hungary, 2003, pp.1-9.

[2] Priestley, M., “DITA XML: A Reuse by Reference Architecture for Technical Documentation”, *In Proceedings of the 19th Annual International Conference on Computer Documentation,* ACM SIGDOC 2001, Santa Fe, New Mexico, USA, October 21-24, 2001. pp. 152-156.

[3] Stojanovic, Lj., Stabb, S. and Studer, R., “eLearning based on the Semantic Web,” *In Proc. of the WWWNet Conf.*, Orlando, USA, 2001.

[4] Verbert, K. and Duval, E., “Towards a global architecture for learning objects: a comparative analysis of learning object content models,” *In Proc. of the 16th ED-MEDIA 2004 Conf.*, Lugano, Switzerland, 2004, pp. 202-209.

[5] R. E. Horn. Structured writing as a paradigm. In Instructional Development: the State of the Art. Englewood Cliffs, N.J., 1998.

[6] DITA Language Reference. Release 1.2. First Edition, May 2003. http://xml.coverpages.org/DITALangugeReference20030606.pdf.

[7] Bechhofer, S., et al (2004) “OWL Web Ontology Language Reference,” W3C Recommendation, http://www.w3.org/TR/2004/REC-owl-ref-20040210.

[8] Djurić, D, Gašević, D., Devedžić, V., “Ontology Modeling and MDA,” *Journal of Object Technology*, Vol. 4, No. 1, 2005, pp. 109-128.

[9] Buccella, A., Cechic, A. and Brisaboa, N.R. (2003). “An Ontology Approach to Data Integration,” *Journal of Computer Science and Technology*, Vol.3 No.2, pp. 62-68.

[10] Hatala, M. and Richards, G., “Value-added Metatagging: Ontology and Rule-based Methods for Smarter Metadata,” *In M. Schroeder and G. Wagner (Eds.) Rules and Rule Markup Languages for the Semantic Web (RuleML2003)*, LNCS 2876, Springer-Verlag, pp.65-80, 2003.

[11] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. (2004). SWRL: A Semantic Web Rule Language Combining OWL and RuleML, version 0.5 of 19 November 2003. [Online]. Available:http://www.daml.org/2003/11/swrl/rules-all.html

[12] Verbert, K., Jovanović, J., Gašević, D., Duval, E., and Meire, M., “Towards a Global Component Architecture for Learning Objects: a Slide Presentation Framework”, In Proc. Of the *17th ED-MEDIA 2004 Conf.*, Montreal, Canada, 2005. (to appear).

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Goal Transition Model and Its Application for Supporting Teachers based on Ontologies

Toshinobu KASAI[†] Haruhisa YAMAGUCHI[†]
[†]*Faculty of Education,*
Okayama University, Japan
Kazuo NAGANO[††] Riichiro MIZOGUCHI[‡]
[††]*Department of Education,* *University of the Sacred Heart, Japan*
[‡]*The Institute of Scientific and Industrial Research,* *Osaka University, Japan*

Abstract. In Japan, the “Period of Integrated Study” program to enhance practical skills began in elementary and secondary education in 2002. Most goals of this program involve meta-ability, which cannot be fully learned by traditional Japanese instructional methods. For this reason, it is necessary and important to provide instructors with a powerful help system that can locate and provide access to a variety of useful information resources. To this end, we built a system that reconstructs the resources according to various viewpoints based on Semantic Web technology. Further, we propose Goal Transition Model to show a skeleton of the transition of instructional goals based on ontologies. And we propose support functions that are used in the model.

Introduction

In Japanese elementary and secondary education, the acquisition academic knowledge had been regarded as important rather than the enhancement of practical skills. In April 2002, however, the Ministry of Education started the "Period of Integrated Study" program in the elementary and secondary education system. The objective of this program is to cultivate learners' ways of learning and thinking and an attitude of trying to creatively solve problems by themselves. However, because Japanese teachers have little experiences with instruction in practical skills, they lack the specific skills for instructional design. In particular, teachers do not have skills in information technology (IT) education.

As a result of the widespread use of the Internet and the development of numerous large information systems, the necessity and importance of IT education have increased. However, there are very few specialist teachers who have the specific skills for teaching IT. Further, it is difficult for them to gain the necessary knowledge and skills, since the educational goals and techniques of IT instruction are not yet clearly defined. For example, most of the teachers who are not specialists mistakenly believe that the use of the technology itself is the main goal of IT education, though the ability to use information systems is a more complex and indispensable aspect of IT education.

Many organizations provide web pages that provide various useful resources for teachers--e.g., digital content, lesson plans, and Q&A [1], [2]. However, it is very difficult to collect the necessary resources for teachers because the relevant web pages are too numerous, and their formats and viewpoints are not unified even when the resources have the same purpose.

One cause of these problems is that various concepts related to IT education and practical skills are not yet clearly defined. Because most of the guidelines and commentaries about the "Period of Integrated Study" present the concepts in a disorganized fashion, we believe that these concepts are not conveyed to teachers effectively. To solve this problem, it is necessary to clarify and articulate the fundamental concepts of practical skills. We believe that ontological engineering can assist in meeting this goal. The ontology provides a common vocabulary/concepts and fundamental conceptual structure of IT education and can promote the reuse and sharing of these concepts among teachers. However, because the ontology is quite abstract, we think that it is not effective to directly provide teachers with it. So, in this study, we use the ontology as a basis and introduce educational goals for practical skills to define other useful information. If useful web resources for the "Period of Integrated Study" are tagged on the basis of ontology, they can be accessed according to the various viewpoints they might have. This framework is realized based on Semantic Web technology.

One of the authors reports on [4] a classification of the goal of IT education in the "Period of Integrated Study" in terms of those which are familiar to the teachers and explains the resource. Although the terms have been well accepted by teachers, they need quite a few modifications from the ontological engineering viewpoint. We make use of the results of this research by identifying the relations between this ontology and our ontologies. Our method is compliant with the openness of the Semantic Web in that it allows the alignment of separate ontologies. Further, we propose Goal Transition Model that shows an skeleton of the transition of the instructional goals based on ontologies. If the skeleton of each provided lesson plan are expressed based on this model, teachers can judge whether or not the plan is appropriate for their instructional objectives without reading it in detail. In this paper, we propose support functions for them which are used in the model.

1. An Outline of Our Approach That Complies with the Openness of the Semantic Web

In this section, we describe the framework for realizing a system that provides teachers in elementary and secondary education with useful resources in accordance with the various viewpoints that they might have. This framework is an example of the Semantic Web application system that is open to the decentralized world. An outline of this framework is shown in Figure 1.

This framework includes two instances of Semantic Web components: one is based on our ontologies, which is described later in detail. We authored metadata of various resources about IT education and the "Period of Integrated Study" in RDF using the ontology of the goal of IT education and the ontology of the fundamental academic ability as the tag; the other Semantic Web component is based on the Goal List of IT education, which was taken from the other research result [4].

The purpose of the Goal List is to provide teachers with teacher-friendly terms by which they can easily express and evaluate the learner's activity during IT instruction in the "Period of Integrated Study" program. Because the Goal List was not generated based on the ontology theory, its quality is not as high as that of an ontology [5]. However, the Goal List already has been so widely used for annotation of large number of information resources of IT education in Japan with the same purpose as an ontology. Therefore, in this paper, we regard this Goal List as an ontology.

In this study, we realize semantic integration between the metadata based on separate ontologies by describing the relations between our ontologies and the Goal List clearly. For example, in this framework, the system can reconstruct lesson plans tagged on the basis of the Goal List from the viewpoint of our ontologies and provide them with it. In addition, the system can integrate lesson plans based on the Goal List with digital contents based on our ontologies which can be used in each step in the lesson plans. This framework enables teachers to use many useful resources more effectively for a wider range of purposes.

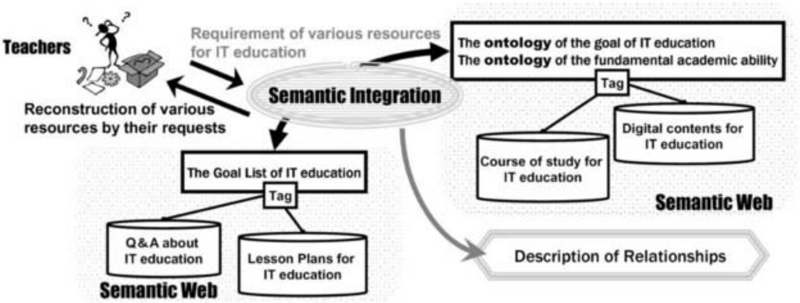

Teachers
Requirement of various resources for IT education
Reconstruction of various resources by their requests
Semantic Integration
The ontology of the goal of IT education
The ontology of the fundamental academic ability
Tag
Digital contents for IT education
Course of study for IT education
Semantic Web
The Goal List of IT education
Tag
Q&A about IT education
Lesson Plans for IT education
Semantic Web
Description of Relationships

Figure 1. The outline of our approach that is compliant with the *openness* of the Semantic Web

2. Our Two Ontologies and Relationships Between These And the Goal List

2.1 The Ontology of the Goal of IT Education

We have built the ontology of the goal of IT education [5]. In this paper, we do not explain this ontology in detail due to space limitation, but we explain only the outline of this.

The ontology of the goal of IT education should consist solely of the goal concepts. Stratification based on an *is-a* relation has to reflect the essential property of these concepts, and ensures that no confusion of various concepts occurs; such confusion can obstruct teachers' understanding of the concepts of IT education. For this ontology, we extracted three concepts that can be the goal of IT education. These are "Knowledge about information/IT", "Skills to use it in the information society", and "Independent attitude in the information society". This classification is compliant with Bloom's taxonomy of instructional objectives [6]. Furthermore, we classified these three concepts into finer classes (subgoals).

2.2 The Ontology of Fundamental Academic Ability

For elementary and secondary education, the Ministry of Education determined a Courses of Study that cultivates a "zest for living," i.e. the ability to learn and think independently, as well as the acquisition of rudiments and basics. For that purpose, the "Period of Integrated Study" was created to cultivate learners' ways of learning and thinking and an attitude of trying to creatively solve or pursue problems by themselves. We extracted and classified goals of the "Period of Integrated Study" as ontology of the fundamental academic ability. This ontology is shown in Figure 2. For this ontology, we classified three concepts, namely "Knowledge to live in the society", "Skills to live in the society" and "Independent attitude in the society", similar to the goal of IT education.

We can regard the "Ability to utilize information," which is the whole of goal of IT education, as a specialized area of fundamental academic ability that is necessary in the information society. Here, we clarify the boundary between the goal of IT education and fundamental academic ability. We define all of the concepts involved in the ontology of the goal of IT education as "academic ability," which is necessary to utilize digital information under an environment based on the information system and the information and telecommunications network. For example, "Skill to investigate," which is one of the concepts of the ontology of fundamental academic ability, means a skill to get necessary information (including non-digital information). On the other hand, a skill to get necessary digital information using IT is "Skill to collect information through IT," which is one of the concepts of the ontology of the goal of IT education. Some pairs, as in this example, exist in two ontologies. A relation of these pairs is that the specialized concept of the ontology of the fundamental academic ability is the concept of the ontology of the goal of IT education. And, this specialization means that an object of a concept of the ontology of the fundamental academic ability is specialized into digital information.

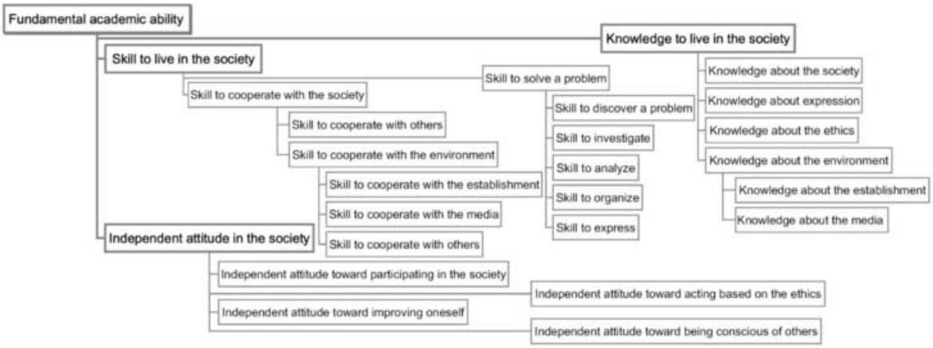

Fundamental academic ability
Knowledge to live in the society
Skill to live in the society
Skill to solve a problem
Skill to cooperate with the society
Skill to cooperate with others
Skill to cooperate with the environment
Skill to cooperate with the establishment
Skill to cooperate with the media
Skill to cooperate with others
Skill to discover a problem
Skill to investigate
Skill to analyze
Skill to organize
Skill to express
Knowledge about the society
Knowledge about expression
Knowledge about the ethics
Knowledge about the environment
Knowledge about the establishment
Knowledge about the media
Independent attitude in the society
Independent attitude toward participating in the society
Independent attitude toward acting based on the ethics
Independent attitude toward improving oneself
Independent attitude toward being conscious of others

Figure 2. The ontology of the fundamental academic ability

2.3 Description of Relationships between Our Ontologies and the Goal List of IT Education

The concepts in the two ontologies that we built do not show practical skills but rather necessary fundamental skills in practice. In other words, they are concepts of high generality which can be applied in various situations. However, it is difficult for teachers to make sense of such concepts of high generality and to make use of these in instructional design. It is therefore necessary to describe the relationships between these concepts and practical activities that cultivate practical skills.

In the Goal List of IT education, for this purpose, examples of concrete learning activities that are easy for teachers to understand are provided together with information that shows when learners should attain this goal. Each example of these learning activities is practical and contains educational goals. We authored metadata related to these learning activities which belong to the respective concepts of the Goal List in RDF. We authored them using the vocabularies defined in the RDF-Schema related to the concepts of the ontology of the goal of IT education and fundamental academic ability. Thanks to this description, the system, which is the Semantic Web application, can reconstruct lesson plans tagged based on the Goal List from the viewpoint of our two ontologies.

3. The Problem-Solving Process and the Goal Transition Model

As mentioned in the above, the concepts in our two ontologies are those of high generality which can be applied in various situations. If more concrete situation of activity is fixed, these concepts of educational goal are set with a role in the situation in detail according to at the concreteness level of abstraction. The most concrete activities are actual learning activities in an actual class. Though there are various ways to make situations more concrete, in this paper, we mainly investigate the situation where a purpose of learning activities is problem-solving (parts of the problem-solving process), since the "Period of Integrated Study" program makes much of cultivating the ability to solve various problems in society. Next, we explain a general process for problem-solving and describe the fundamental academic ability that is necessary in each step of this process as educational goal.

3.1 The Problem-Solving Process and the Educational Goals That are Necessary at Each Step

In this study, we defined the Problem-Solving Process which is more general as a cycle shown in the left figure in Figure 3 with referring to National Geography Standards [11]. The educational goals at each step of this process are extracted from our two ontologies. These are shown in the right figure in Figure 3.

Each concept of these educational goals has a role in this process. For example, although "Skill to analyze" appears in two different steps, the roles in the problem-solving process are different from each other. Its role in the step of "Classification, analysis and judgment" involves the analysis of various kinds of information (including non-digital information) collected to solve the problem. Its role in the step of "Self-evaluation" involves the analysis to evaluate the process of problem-solving by oneself. The concepts of academic ability are necessary in steps of the problem-solving process, and these concepts have a leading role in the process. In this paper, we call these concepts "leading skills" in the problem-solving process. And in this process, if a more concrete activity is given in each step, other concepts of academic ability are set with more detailed roles.

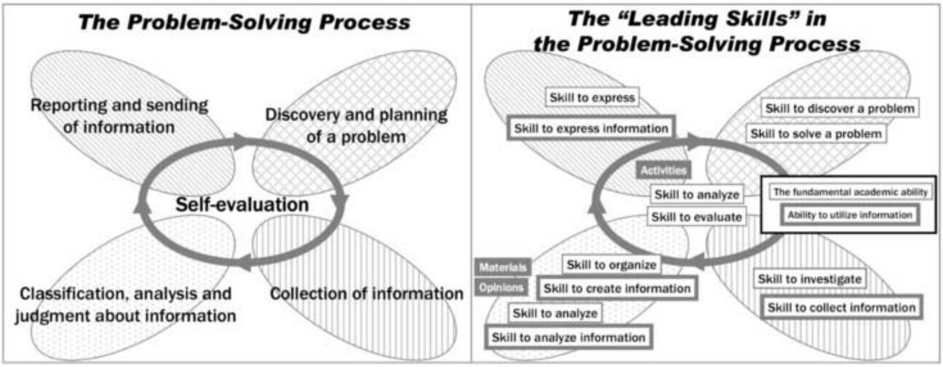
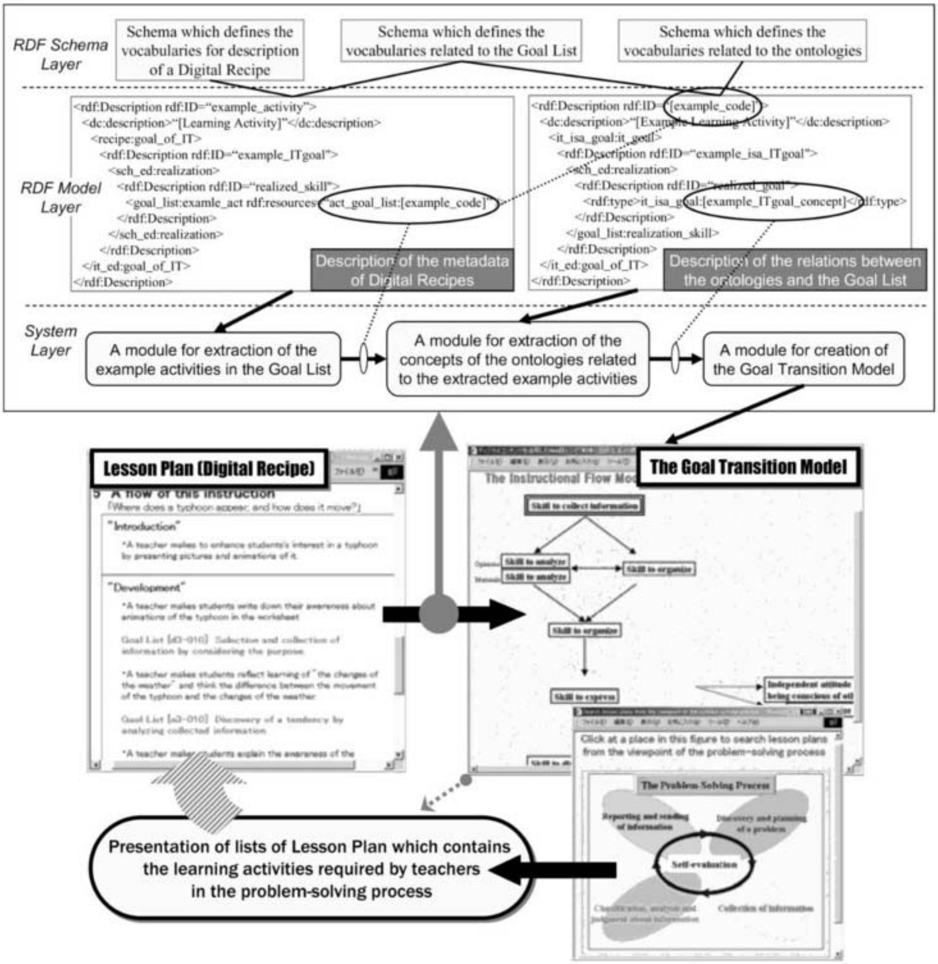

The Problem-Solving Process
Reporting and sending of information
Discovery and planning of a problem
Self-evaluation
Classification, analysis and judgment about information
Collection of information
The "Leading Skills" in the Problem-Solving Process
Skill to express
Skill to express information
Skill to discover a problem
Skill to solve a problem
Activities
Skill to analyze
Skill to evaluate
The fundamental academic ability
Ability to utilize information
Materials
Opinions
Skill to organize
Skill to create information
Skill to analyze
Skill to analyze information
Skill to investigate
Skill to collect information

Figure 3. The problem-solving process and the leading skills in this process

3.2 The Goal Transition Model

Most lesson plans of the "Period of Integrated Study" program which are provided via the internet aim to cultivate practical skills to be used in the problem-solving process. If all of the leading skills of the problem-solving process are extracted in order from each lesson plan, it is possible to express a skeleton of the instruction from the perspective of the problem-solving process. In this study, we call this skeleton "the Goal Transition Model". All concepts which can be used in this model are defined in our two ontologies. An example of a Goal Transition Model extracted from an actual lesson plan is shown on the right at the center in Figure 4.

Here, "Skill to analyze," which exists in different steps of the problem-solving process, can be distinguished by considering its role. In this study, we classify and describe objects of analysis clearly to judge which step it is. The object of "Skill to analyze" in the step of "Classification, analysis and judgment" is "materials" or "opinions" because its role is the analysis of various kinds of information collected to solve the problem. The object of "Skill to analyze" in the step of "Self-evaluation" is "activities" because its role is the analysis to evaluate the process of problem-solving performed by learner's self. In this study, we use "problems", "learner's self", "others" and "situation" as objects of analysis in addition to the three objects mentioned above, "materials", "opinions" and "activities". However, "Skill to analyze" is regarded as a leading skill in the problem-solving process only when its object is one of these latter three objects. Otherwise, this concept is regarded as simply another goal concept. In the Goal Transition Model, the other concepts are connecting to the side of the "leading skill," which is contained in the same learning activities as shown on the right at the center in Figure 4.


RDF Schema Layer
Schema which defines the vocabularies for description of a Digital Recipe
Schema which defines the vocabularies related to the Goal List
Schema which defines the vocabularies related to the ontologies
RDF Model Layer
Description of the metadata of Digital Recipes
Description of the relations between the ontologies and the Goal List
System Layer
A module for extraction of the example activities in the Goal List
A module for extraction of the concepts of the ontologies related to the extracted example activities
A module for creation of the Goal Transition Model
Lesson Plan (Digital Recipe)
The Goal Transition Model
The Instructional Flow Mo
Skill to collect information
Skill to analyze
Skill to analyze
Skill to organize
Skill to organize
Skill to express
Independent attitude
being conscious of oth
The Problem-Solving Process
Reporting and sending of information
Discovery and planning of a problem
Self-evaluation
Classification, analysis and judgment about information
Collection of information
Presentation of lists of Lesson Plan which contains the learning activities required by teachers in the problem-solving process

```
<rdf:Description rdf:ID="example_activity">
 <dc:description>"[Learning Activity]"</dc:description>
  <recipe:goal_of_IT>
   <rdf:Description rdf:ID="example_ITgoal">
    <sch_ed:realization>
     <rdf:Description rdf:ID="realized_skill">
      <goal_list:examle_act rdf:resources="act_goal_list:[example_code]">
     </rdf:Description>
    </sch_ed:realization>
   </rdf:Description>
  </it_ed:goal_of_IT>
</rdf:Description>
```

```
<rdf:Description rdf:ID="[example_code]">
 <dc:description>"[Example Learning Activity]"</dc:description>
  <it_isa_goal:it_goal>
   <rdf:Description rdf:ID="example_isa_ITgoal">
    <sch_ed:realization>
     <rdf:Description rdf:ID="realized_goal">
      <rdf:type>it_isa_goal:[example_ITgoal_concept]</rdf:type>
     </rdf:Description>
    </goal_list:realization_skill>
   </rdf:Description>
  </it_ed:goal_of_IT>
</rdf:Description>
```

5 A flow of this instruction
「Where does a typhoon appear, and how does it move?」
"Introduction"
*A teacher makes to enhance students's interest in a typhoon by presenting pictures and animations of it.
"Development"
*A teacher makes students write down their awareness about animations of the typhoon in the worksheet
Goal List [a3-010] Selection and collection of information by considering the purpose
*A teacher makes students reflect learning of "the changes of the weather" and think the difference between the movement of the typhoon and the changes of the weather
Goal List [a3-010] Discovery of a tendency by analyzing collected information
*A teacher makes students explain the awareness of the

Click at a place in this figure to search lesson plans from the viewpoint of the problem-solving process

Figure 4. Two functions which support teachers by using the Goal Transition Model

4. Building of a Support System for Teachers by Using the Goal Transition Model

We have built the support system including functions that are realized by using the Goal Transition Model based on the framework which explained in Section 1. In this section, we describe how to create this model from lesson plans and these two implemented functions.

4.1 How to Create the Goal Transition Model from a Lesson Plan

The resources used by this system are simple lesson plans on the Web (called Digital Recipes) [2] provided by Okayama Prefecture Information Education Center. These Digital Recipes are open to the public as resources related to concepts of the Goal List. However, they were not described as metadata in the Semantic web sense. So we authored the metadata of these resources from the viewpoint of the Goal List. A procedural flow to create the Goal Transition Model from the metadata of a Digital Recipe by the system is shown at the top in Figure 4.

The system analyzes the metadata of a Digital Recipe we produced and extracts concepts of the Goal List tagged in this resource, and then the system extracts the concepts of the ontology of the goal of IT education and the ontology of the fundamental academic ability related to those concepts of the Goal List from the other resource (this describes the relations between our two ontologies and the Goal List). Next, the system connects and outputs the leading skills in the order of the problem-solving process. Further, the system outputs each other concept at the right side of the leading skill contained in the same learning activity that contains it. Here, when the different concepts which are in the same step of the problem-solving process and are repeated, the system outputs these concepts in parallel from the previous leading skill. This is because these concepts which are in the same step cannot be arranged.

4.2 Details of Implemented Functions by Using the Goal Transition Model

One function builds the Goal Transition Model of a lesson plan (Digital Recipe) automatically and provides teachers with it as shown at the top in Figure 4. For this function, teachers can get the skeleton of this lesson from the viewpoint of educational goals without going through the lesson plan in detail. This skeleton provides teachers with the true nature of the lesson, which can be difficult to uncover among superficial information such as learning activities, information systems, digital contents and so on. Therefore, we think that this function is useful for teachers who are not accustomed to the cultivation of practical skills.

The other function searches necessary lesson plans from the viewpoint of the problem-solving process according to requirement of teachers. By clicking on the place which shows each step in the problem-solving process, teachers can get lists of lesson plans which contain the learning activities required as shown at the bottom in Figure 4. In Japan, although IT education and the "Period of Integrated Study" program attach importance to the cultivation of an ability to solve problems, the function which can search the necessary lesson plans which are open to the public from the viewpoint of a step in the problem-solving process is nearly nonexistent. In this study, this function is realized by using the framework of the Semantic Web based on ontologies and the Goal Transition Model that we proposed.

4.3 Evaluation of Our Approach

We have evaluated the effectiveness of the ontology of the goal of IT education by an experiments with 21 high school teachers [3]. In this evaluation, it was shown both qualitatively and quantitatively that our ontology is effective on deepening teachers' understanding of the goal of IT education. And, it was shown that teachers had two kinds of opinions about the use of the ontology: One is that the presentation of the ontology itself is not very helpful for teachers to design better instruction of IT education and the other is that the addition of the ontology to the other support resources enhances the utility of its resources for teachers. But, we have not evaluated the proposed Goal Transition Model and its application function yet. In the near future, we intend to evaluate them.

5. Related Work

Many organizations and researchers have been trying to enhance shareability and reusability of various educational resources. Here, we introduce some of these efforts that are related to our approach briefly.

The Learning Object Metadata (LOM) was provided by The IEEE Learning Technology Standards Committee (LTSC) [8]. The LOM specifies the syntax and semantics of Learning Object Metadata, defined as the attributes required to full/adequate description of a learning object. We cannot describe the contents of the Learning Objects in compliance with the LOM standards because they focus on the minimal set of attributes to allow these LOs to be managed, located, and evaluated in total independence of their contents. Our approach of this paper aims at describing the contents by limiting objects to lesson plan.

There is the IMS Learning Design project which aims at making the standard to describe the instruction/learning activities, the learning environment, and the learning objectives that can be expressed in lesson plan [7]. In compliance with this standard, we can express the contents of lesson plan in detail. However, we think that this expression is too complex for teachers who do not understand the contents and goal of education enough yet. Our approach aims at expressing them with solely educational goal for the teachers who do not understand them.

And, there are some researches based on these standards and various ontologies [9], [10]. The goal of [9] is to specify an evolutional perspective on the Intelligent Educational Systems (IES) authoring and in this context to define the authoring framework EASE: powerful in its functionality, generic in its support of instructional strategies and user-friendly in its interaction with authors. And, the study [10] proposes a theory-aware ITS authoring system based on the domain and task ontologies of instructional design. We intend to build a support system for designing an instructional system for cultivating practical skills to solve various problems based on the framework which is proposed in this paper with referring to the results of these related works.

6. Summary

In this paper, we described two ontologies; the ontology of the goal of IT education and the ontology of the fundamental academic ability. And, we proposed a framework to make use of the results of another research [4] by alignment of these ontologies based on Semantic Web technology. Further, we proposed a Goal Transition Model that shows a skeleton of the transition instructional goals from a lesson plan, and a support system that has functions realized by this model.

References

[1] The Meeting of Tuesday (2002), The curriculum lists of information education in the "Period of Integrated Study", HomePage of the Meeting of Tuesday, http://www.kayoo.org/sozai/.

[2] Okayama Prefectural Information Center (2002), Okayama Prefectural Information Education Center, Digital Contents Recipes and Worksheets, http://www2.jyose.pref.okayama.jp/cec/webresipi/index.htm.

[3] T. Kasai, H. Yamaguchi, K. Nagano, R. Mizoguchi (2005), Systematic Description of the Goal of IT Education Based on Ontology Theory, IEICE Trans. on Information and Systems, J88-D-I, No.1, pp.3-15.

[4] The Meeting of Tuesday (2001), The Meeting of Tuesday, The Goal List of Information Education, Mail-Magazine of the Meeting of Tuesday, http://kayoo.org/home /project/list.html.

[5] T. Kasai, H. Yamaguchi, K. Nagano, R. Mizoguchi (2004), Building of an Ontology of the Goal of IT Education and Its Applications, In Proceedings of the 3rd International Workshops of Applications of Semantic Web Technologies for E-Learning (SW-EL'04), pp.55-65.

[6] B.S. Bloom, J. T. Hastings, G. F. Madaus (1971), Handbook on formative and summative evaluation of student learning, McGraw-Hill.

[7] IMS (2002), IMS Learning Design Specification ver.1.0, http://www.imsglobal.org/learningdesign/.

[8] IEEE LTSC (2002), IEEE Standard for Learning Object Metadata, http://ltsc.ieee.org/wg12/.

[9] L. Aroyo, A. Inaba, L. Soldatova, R. Mizoguchi (2004), EASE: Evolutional Authoring Support Environment, Proc. of the seventh International Conference on Intelligent Tutoring Systems (ITS2004).

[10] J. Bourdeau, R. Mizoguchi (2000), Collaborative Ontological Engineering of Instructional Design Knowledge for an ITS Authoring Environment, Proc. of the 6th International Conference on Intelligent Tutoring Systems (ITS2002), pp.399-409.

[11] Geography Education Standards Project (1994), Geography for Life: National Geography Standards, National Geographic Research and Exploration.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Exploiting Readily Available Web Data for Scrutable Student Models

Judy KAY and Andrew LUM
School of Information Technologies,
University of Sydney, NSW, 2006, Australia

Abstract. This paper describes our work towards building detailed *scrutable* student models to support learner reflection, by exploiting diverse sources of evidence from student use of web learning resources and providing teachers and learners with control over the management of the process. We build upon our automatically generated light-weight ontologies using them to infer from the fine-grained evidence that is readily available to higher level learning goals. To do this, we have to determine how to interpret web log data for audio plus text learning materials as well as other sources, how to combine such evidence in ways that are controllable and understandable for teachers and learners, as required for scrutability, and finally, how to propagate across granularity levels, again within the philosophy of scrutability. We report evaluation of this approach. This is based on a qualitative usability study, where users demonstrated good, intuitive understanding of the student model visualisation with system inferences.

1. Introduction

Student models have one obvious role as the drivers for personalisation [1]. Importantly, externalised or open student models have another invaluable potential role, to help learning by enabling improved learner reflection [2]. They also can be a useful basis for feedback to the teachers [3]. We would like teachers to enhance their web-based or web-enhanced courses with learner models useful for reflection. This means that the processes of building the learner models need to be tailored to typical classroom teachers, being understandable and quick to use. To make the models useful for reflection, they must model the learners at varying levels of granularity: coarse grained so that learners can see how they are doing on the overall learning goals; and fine grained so that they can determine which elements of work contribute to this higher level goals [5]. Moreover, we want learners, and teachers, to feel in control of the modelling and to be able to scrutinise the models, delving into details of the processes that determine the model.

Web-based and other interactive learning systems differ from typical classroom learning in that they can easily provide very large amounts of data about learner. Unfortunately, that data is typically of very poor quality, as for example, in the case of detailed logs of page visits, time spent on each page and links selected. These give weak evidence that the user read the material, let alone learnt it. On the other hand, learners who have never visited the web pages for a course are unlikely to have learnt the course material. Evidence of this sort is so readily available that it would be valuable to exploit it to build student models. Web learning environment also may provide higher quality evidence about learners. For example, there may be marks for class exercises, results of on-line quizzes and multiple choice questions. Such evidence tends to be fine grained in the sense that a single page of an on-line course is about a small part of it and a quiz question or set is typically about a current, specific sub-topic [9].

We want to support scrutable learner modelling which exploits the combination of the full range of types of evidence available. This poses several challenges. First, we need to determine *how to interpret the evidence available*. For example, we have a course with on-line lectures, each composed of a series of text slides with audio content. We need to determine how to interpret the evidence that a student *attended* such a lecture. Secondly, once the evidence is available, we need to combine diverse evidence sources, a task that has been the subject of a substantial body of research, including for example [10, 11, 12, 13] We want this process to be readily controllable by learners and teachers and to be scrutable so that our system can provide simple explanations about how the modelling works. A third problem is to be able to *reason from the fine-grain level* of the available evidence to the coarser grained higher level concepts. Our approach exploits an existing tool, Mecureo [6], which builds light-weight ontologies automatically by analysing subject-area glossaries. This approach is very attractive in relation to our goal of scrutability because the dictionary is then a useful resource for explanations of the ontology: we can simply explain why the system treats two concepts as related by showing the relevant dictionary definitions. This approach also meets our goals of low cost construction of student models since it defines a structure for the user model automatically. There is much work on ontological inference using formal specifications and axioms such as [14] but cannot operate on our light-weight Mecureo-generated ontologies.

This paper describes our work towards tackling these challenges. Section 2 outlines our approach and Section 3 discusses the evaluation framework and infrastructure. Section 4 presents the results of a user study and Section 5 concludes with related work and discussions.

2. Reasoning from readily available evidence to student models

We have identified three important steps to reason about the available evidence in the ontology:

1. define how the available data contributes to the student model;
2. combine available evidence for a component concept;
3. reasoning about the high level concepts.

The student model shown in Fig. 1. illustrates how evidence feeds mainly into the fine-grain concepts. Evidence may feed into a single concept (E1, E3 and E5) or multiple (E2 and E4). Evidence may also feed into higher level (non-leaf) concepts of the ontology (E4). They may also come from different sources (E1, E3 and E5 are from web log data; E2 and E4 are from tutorial marks). The higher level concepts, *Usability* and *Predictive* have no direct evidence sources.

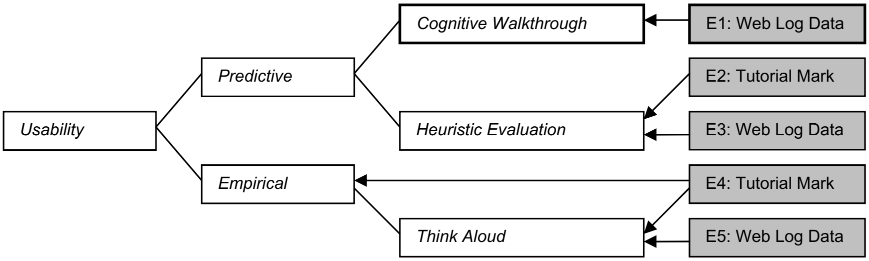


Fig. 1. A student model with fine-grain evidence for learner knowledge of concepts in the HCI domain. It shows the coarse grain concept *Usability* on the left, with finer grain subsumptive topics to the right. Evidence feeds into the finest grain concepts.

To tackle these problems of varying quality of evidence from different sources and varying amounts of evidence, we introduce the notion of a Student Standard. Using a comparison to the Standard Student model we end up with a relative measure rather than an absolute one, reducing the effect of the varying amounts of evidence for the concepts. In the case of a course or teaching system, the Student Standard may be defined as the teacher sees fit: for example, a teacher in a mastery-based course may define it as the student model of the student who earns full marks for assessments and a perfect attendance record by the end of the course. In Fig. 1. we can consider the case where a "bare-pass" standard where the student is not required to visit the web pages for *Cognitive Walkthrough* (highlighted with a bold border in the figure), whereas a "advanced student" standard does. This is similar to overlays in [4] except that there is no single expert model, rather one or more models the teacher considers meaningful.

Consider the student model shown in Fig. 1. with two types of evidence: the amount of time students spent listening to audio for online learning objects mined from web log data, and the marks they received for weekly tutorial sessions. We take the Standard Student as the student that attains full marks in the tutorials and listens to all the audio on the online lectures.

Step 1.

For the audio evidence, the length of audio narrative for each slide is known. We assume the Standard Student will have listened to the full slide (and have an extra bit of leeway time for taking notes, etc). We can compare the length of time a user has spent on each slide to that of the Standard Student time, and assign a score based on this. The weightings we assign range from 0.0 to 1.0 and the breakdown are shown in Table 1.

Table 1. Understanding of audio slides based on duration stayed

Understanding	**Duration on slide as percentage of Standard Student Time**	**Weighting**
Seen	Student Time < 10%	0.1
Partial Heard	10% <= Student Time < 80%	0.5
Standard Student	80% <= Student Time < 150%	1.0
Overheard	150% <= Student Time	0.8

The Overheard weighting is slightly lower than the Full Heard. This is to account for the times when students have become distracted with other activities and have left the browser open. All of the values from each audio evidence source for a concept are then averaged. This results in a final value from 0 to 1.0; a perfect student will have listened to every slide as a Full Heard, resulting in a value of 1.0 for the component. We call this the Normalised Audio Score.

For the tutorial evidence, the students receive a mark out of 10. A perfect student should get full marks for every tutorial in our course, so in effect a mark out of 10 is already a comparison against that of the Standard Student. We sum all the tutorial evidence scores for a particular concept and divide by the total possible marks (Standard Student's score) to get a value between 0.0 and 1.0 for the final value for tutorial evidence. We call this the Normalised Tutorial Score.

Step 2.

To combine the two values, we use a simple formula to determine each evidence type's contribution to the final score:

$$\text{Score} = k1*(\textit{Normalised Audio Score}) + k2*(\textit{Normalised Tutorial Score}) \text{ where } (k1 + k2) = 1. \quad (1)$$

Based on an intuitive sense of the reliability, k1 has been set to 0.25, and k2 has been set to 0.75 when there is tutorial evidence. This formula can be easily generalised to any number of evidence sources.

Step 3.

We need to be able to model about the user's knowledge of higher level concepts. We want to deal with the case where there is no direct evidence at all. For example, in Fig. 1., there is no direct evidence for the concept *usability* as evidence sources contributes to concepts finer grain.

One simple method is to do a spanning tree from the leaf concepts (the fine grain) and recursively pass their values up to the parent concept till we reach the higher level coarse grain concept we want to reason about. At each stage when the values are passed up the tree, some calculations can be done to factor in the distance from the course grain concept in the tree, as well as the amount or type of evidence.

An example of this is the averaging model we present below. We can recursively run this algorithm up the tree till we reach the root concept we are inferring about.

For a particular concept v_a, we take an average of the values of the child concept values $\{v_{a,1},.., v_{a,n}\}$. This value is then multiplied with (1 - value of root concept) and added to the value of the root concept to give a proportional boost, but always maintaining a value between 0 and 1. The lower the score of the root concept, the higher proportion of inference the value will take. Equation (2) summarises the averaging formula for a concept v_a with n related concepts, where $n >= 1$. In the case of $n = 0$, $v_a' = v_a$ (i.e. there is no inferred contribution to the final score for this concept).

$$v_a' = v_a + (1 - v_a) * (\frac{1}{n} \sum_{v_{a,i} \in v_{a.child}} v_{a,i}) \text{ where } v_{a.child} = \{v_{a,1}, \ldots, v_{a,n}\} \quad (2)$$

Consider the example portion of a student model shown in Fig. 1. We want to infer about concept *Predictive*. Assume the two related sub-concepts *Cognitive Walkthrough* and *Heuristic Evaluation* have values of 0.6 and 0.4 respectively, and *Predictive* has a value of 0.1. We substitute these values into formula (2) and arrive at the value 0.65 as the new value for *Predictive* – a quite reasonable assumption based on the knowledge of the fine grain concepts (3 & 4).

$$v_{predictive}' = v_{predictive} + (1 - v_{predictive}) * \frac{v_{cognitive\ walkthrough} + v_{heuristic\ evaluation}}{2} \quad (3)$$

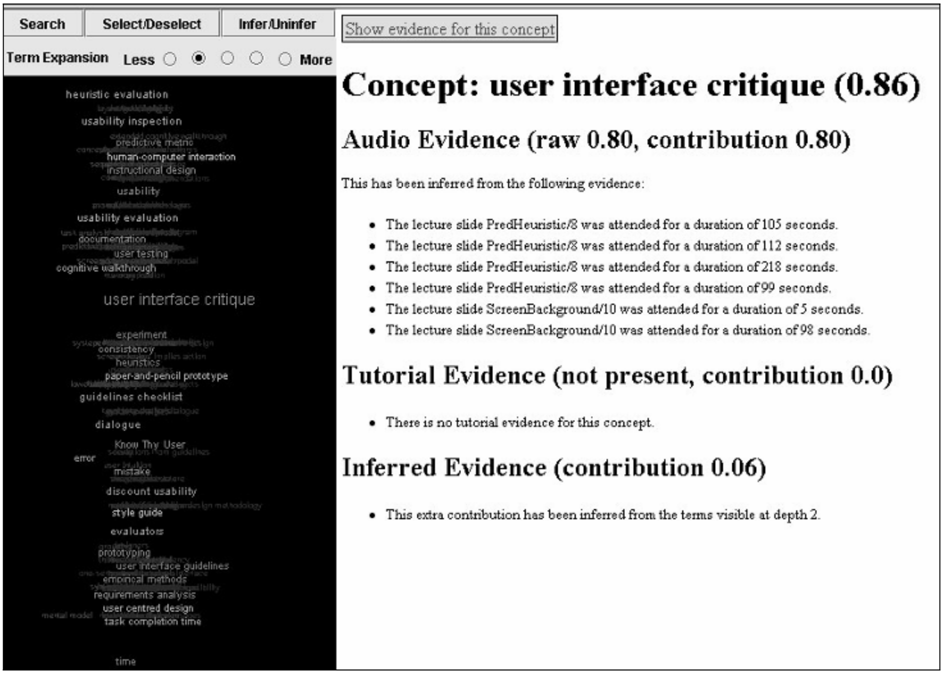
$$v_{predictive}' = 0.1 + (1 - 0.1) * \frac{0.6 + 0.4}{2} = 0.65 \quad (4)$$

3. Evaluation Framework

The *User Interface Design and Programming* course taught at this university is the demonstration environment for the tools and also the evaluation domain. It has 241 audio-slides (lectures are a collection of visual slides with audio narrative). There are also live lectures and laboratory classes. For the evaluation, we used the subset of material about design and HCI (161 slides organised in 9 lectures).

We now describe, very briefly, the process used to build the student models. This draws upon several tools that we have constructed:

- Mecureo [6] to construct the domain ontology;
- Metasaur [7] to link each learning object with metadata concepts from the ontology;



Concept: user interface critique (0.86)

Audio Evidence (raw 0.80, contribution 0.80)

This has been inferred from the following evidence:

- The lecture slide PredHeuristic/8 was attended for a duration of 105 seconds.
- The lecture slide PredHeuristic/8 was attended for a duration of 112 seconds.
- The lecture slide PredHeuristic/8 was attended for a duration of 218 seconds.
- The lecture slide PredHeuristic/8 was attended for a duration of 99 seconds.
- The lecture slide ScreenBackground/10 was attended for a duration of 5 seconds.
- The lecture slide ScreenBackground/10 was attended for a duration of 98 seconds.

Tutorial Evidence (not present, contribution 0.0)

- There is no tutorial evidence for this concept.

Inferred Evidence (contribution 0.06)

- This extra contribution has been inferred from the terms visible at depth 2.

Fig. 3. Example SIV interface[1]. The visualisation is at the left, with the concepts listed vertically. The concept *user interface critique* is in focus and has the largest font. Related fonts are in the next largest fonts, and unrelated concepts are blurred out. Horizontal position indicates the amount of evidence for that concept in the user model. Concepts with a score greater than 0.5 are in green, others in red. The list of evidence contributing to the concept score is at the right – in this case there is no tutorial evidence, and the score for the concept is 0.86. The inferred evidence is determined using the averaging formula (2).

- Personis [8] to represent the student models;
- slide-evidence extractor which analyses web log data to create evidence based on student accesses to the slides;
- tutorial evidence extractor which uses tutorial marks to created evidence;
- SIV (Scrutable Inference Viewer) [6, 7] to provide the interface for users in the study.

The domain ontology built by Mecureo[6] was automatically from the Usability First Glossary[2]. This has 1,129 terms and categories. Mecureo analysed the dictionary definitions to construct a light-weight ontology based on the relationships between concepts defined in the dictionary. We augmented these with 105 additional definitions, giving a total of 1,234 concepts and 10,690 relationships between them.

We used the Metasaur interface to create metadata, based on the set of concepts in the domain ontology, associating these with each lecture-slide and tutorial [7]. We annotated the first slide of each learning object with high level concepts, and omitted any hits to these pages when analysing the web log data as the slide itself only showed the title of the learning topic.

The subset of the ontology' concepts used in the metadata automatically defined the components of the student model definition in the Personis user modelling representation [8] containing a total of 190 concepts with 423 relationships between them. The tools that

[1] Colour screenshot at http://www.it.usyd.edu.au/~alum/assets/screenshots/siv-um05-1.jpg

[2] http://www.usabilityfirst.com/glossary/main.cgi

collected evidence from web accesses and tutorial performance were used to add evidence to each student's learner model.

The reasoning methods described above operate as *resolvers* in Personis. The result of this process is available for the learner to scrutinise, with the Scrutable Inference Viewer [6, 7] (SIV) interface. This provides an interface for visualizing the user model and to scrutinise the basis for what us displayed. Fig. 3 has a screenshot and explanation of its elements. The 190 concept are displayed in the visualisation, colour overlays give an indication of the student score for that concept.

4. Usability Study

We used a think-aloud evaluation. Participants were six senior level undergraduate students, all with experience as teaching assistants. They were asked to take the role of tutors and were presented with the information sheet below. They were asked to think-aloud as they performed the task in Fig.4. In particular, we were interested in whether they:

1. could use the interface and understand it
2. would consider the results of the inference reasonable
3. could see the related concepts contribute to the reasoning

Two pseudo-students, A and B, were created, both based on a real student at the middle of the class ranking in the User Interface Design and Programming course. They were identical, except that student B had failed to attend several online lectures, and so had no web data for these. In addition, student B had lower tutorial marks than student A. Using SIV inference for course grain concepts, B's scores were consistently lower than A's.

The three concepts, *cognitive modeling*, *heuristics*, and *user interface guidelines* all had no evidence; hence a resolved score of zero in the user model, resulting in bright red font and, as these had no evidence, they were at the far right in the visualisation. Table 2 shows the values for the three concepts after inference based on evidence for related concepts. Student A's higher degree scores for fine grain concepts is also reflected in the inferred values.

Students A and B have quite different competence for the User Interface Design and Programming course. The course coordinator has requested that students struggling in this area will be invited to attend an additional catch-up tutorial session.

As a tutor for the course, you want to see how well the students understand the concepts in the area of predictive usability, in particular the concepts *cognitive modeling*, *heuristics* and *user interface guidelines*. You need to fill out a form to allow them to attend the tutorial session as there is a limited number of places.

Unfortunately there is little direct evidence for these concepts, though there are plenty of more specialized concepts (such as the fact they have listened to a lecture on *cognitive walkthrough*, which is a subtopic of *cognitive modeling*) with evidence that could contribute to their understanding of the concepts you are after.

You want to select these topics on the signup sheet (and maybe some additional ones) relating to this area of study and see what the system infers about the student's knowledge.

Decide if Student A and/or Student B should attend the catch-up tutorial with a justification for why they should attend on the signup sheet.

Fig. 4. The task description for the evaluation given to the participants.

Table 2. The inferred values for each concept

Concept	Student A	Student B
cognitive modeling	0.50	0.22
Heuristics	0.87	0.23
user interface guidelines	0.62	0.33

All the participants successfully completed the task under 10 minutes and from the results in Table 3, unanimously decided that student B should attend the extra tutorial session.

All participants started with the search tool to look for the topics and quickly correlated the colour of the topics with the degree of knowledge for the students. All participants based their judgment student B's poorer understanding compared to student A because student B’s inferred scores were all lower.

Table 3. The information written by participants on the signup sheet

Participant s	Student	Reason for attending extra tutorial session
1	B	They do not have a good understanding of the above 3 concepts.
2	B	Although there is no direct evidence of the student’s understanding of the three concepts, by inferring other concepts that are related to the three concepts, probability of the student understanding the concepts is low.
3	B	Inference readings returned low as no data on many of the related topics.
4	B	Although there is no direct evidence in the form of audio/video evidence of student A or B understanding the concept. The inferred evidence based on the relationships or underlying concepts suggest that student A has more knowledge than student B as the values for the inferred evidence are higher for all three concepts.
5	B	Need more details and info on these topics.
6	B	Low inferred score for all 3. The concepts looked red all the time.

Some pointed out upon seeing student B’s user model that they were not as good as student A based simply on the distribution of the colours when the concepts were expanded. Participant 5 said for the concept *user interface guidelines*, “In this case, there’s more greens for this topic for student A [than student B]”.

They seemed to be happy that the inferred values matched their expectations. Participant 1 selected *cognitive modeling* for student A and instantly said “Cognitive modeling comes up red. I infer because the other concepts are green”. For student B on the same topic, Participant 1 stated “cognitive modelling appears correct [coloured red], but I will infer to make sure”. These comments were made before the participants used the **Infer** button to see the inferred value.

Participants could also correlate the inferred value with the values for related concepts. For example, Participant 6 was asked if they could see why the inferred value for *heuristics* indicated that Student A knew this concept, to which they replied “I guess because all the related stuff is green”.

5. Discussion and Conclusion

Many numerical uncertainty management approaches have been applied to student modelling [10]. However these require a more formal network or ontology structure which differs from our light weight approach.

Our current approach is not without limitations. In this paper we only discuss the reasoning about coarse grain concepts in the case where there is no direct evidence. When there are coarse grain concepts with few (say one or two) sources of evidence, the reliability of the concept's resolved scores is decreased. In future work, we need to consider the amount and type of evidence required by the Standard Student to get a perfect score compared to that of the student.

A second issue is the attributes of the relationship in the ontology. The relationships are (in the case of using Mecureo) not only typed, but also weighted for the strength of the relationship. The formula presented in (2) does not take this into account.

Based on the results of the user study, the approach we propose seems promising. The participants understood the interface and they did consider the results of the inference reasonable. The granularity of the concepts was also realized and the participants could appreciate the fact that reasoning was required about higher level concepts that did not have direct evidence sources.

References

1. Self, J.: The defining characteristics of intelligent tutoring systems research: ITSs care, precisely. In: *International Journal of Artificial Intelligence in Education* Vol. 10. (1999) 350-364.

2. Bull, S.: *Supporting Learning with Open Learner Models*. 4th Hellenic Conference with International Participation: Information and Communication Technologies in Education, Athens. (2004). Keynote.

3. Yacef, K.: Making large class teaching more adaptive with the logic-ITA. In: *Theoretical Proceedings of the sixth conference on Australian computing education* – Vol. 30. ACM International Conference Proceeding Series (2004) 343-347.

4. Carr, B., Goldstein, I.: *Overlays: a theory of modelling for computer aided instruction*. MIT. Cambridge, MA (1977).

5. McCalla, G., Greer, J.: Granularity-Based Reasoning and Belief Revision in Student Models. In Greer, J., McCalla, G. (eds): *Student Modelling: The Key to Individualized Knowledge-Based Instruction. NATO ASI Series, Series F: Computer and Systems Sciences*, Vol. 25. Springer-Verlag, Berlin Heidelberg (1994) 39-62.

6. Apted, T. and Kay, J., MECUREO Ontology and Modelling Tools. In: *WBES of the International Journal of Continuing Engineering Education and Lifelong Learning*. Accepted 2003, to appear.

7. Kay, J., Lum, A.: Building user models from observations of users accessing multimedia learning objects. In: Nuernberger. A, Detyniecki, N (eds): *Adaptive Multimedia Retrieval*, Springer, (2004) 36–57.

8. Kay, J., Kummerfeld, B., and Lauder, P.: Personis: a server for user models. In: *Proceedings of Adaptive Hypertext 2002*. Springer (2002) 203-212.

9. De Bra, P., Calvi, L.: AHA: a Generic Adaptive Hypermedia System. In: *Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia HYPERTEXT'98*, Pittsburgh, USA, (1998) June 20–24.

10. Jameson, A. (1996). Numerical uncertainty management in user and student modeling: An overview of systems and issues. In: *User Modeling and User-Adapted Interaction* Vol. 5 (1996) 193–251.

11. Mislevy, R., Almond, R., Yan, D., Steinberg, L.: Bayes Nets in Educational Assessment: Where the Numbers Come From. In: Laskey K., Prade, H. (eds): *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence* (1999) 437- 446.

12. Conati, C., Gertner, A., Vanlehn, K: Using Bayesian Networks to Manage Uncertainty in Student Modeling. In: *User Modeling and User-Adapted Interaction* Vol. 12. (2002) 371-417

13. Zapata-Rivera, J., Greer, J.: Analyzing Student Reflection in *The Learning Game*. In: Aleven, V. et al (eds): *aied2003 Supplementary Proceedings*. University of Sydney (2003) 288-298.

14. Staab, S., Maedche, A.: Ontology Engineering beyond the Modeling of Concepts and Relations. In: Benjamins R.V., Gomez-Perez A., Guarino N., Uschold M. (eds): *Proceedings of the 14th European Conference on Artificial Intelligence Workshop on Applications of Ontologies and Problem-Solving Methods* (2000).

Acknowledgements: We thank Hewlett-Packard for funding this research.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

What Do You Mean by to Help Learning of Metacognition?

Michiko KAYASHIMA[†], Akiko INABA[‡] and Riichiro MIZOGUCHI[‡]
[†]Department of Human Science, Tamagawa University ‡ISIR, Osaka University
†6-1-1 Tamagawagakuen, Machida, Tokyo, 194-8610, Japan
‡8-1 Mihogaoka, Ibaraki, Osaka, 567-0047 Japan

Abstract. Several computer-based learning support systems and methods help learners to master metacognitive activity. Which systems and methods are designed to eliminate which difficulties associated with the learning of metacognitive activity through its clear specification? We adopt a method in our research that supports learning by eliminating salient difficulties. We believe that it is possible to eliminate or decrease them through appropriate design only after specifying those difficulties associated with learning. In this study, we analyze difficulties in performing cognitive activity, distinguish factors of difficulty from other factors, and construct our framework, which represents difficulties in performing metacognitive activity. Finally, we organize existing support systems and methods based on that framework.

1. Introduction

Several kinds of methods support learning. One kind divides a learning process whose grain size is large into two or more sequential steps, such as getting learner motivated, providing necessary knowledge about it, showing how to use it, and asking the learner to follow these steps in order. Such a method helps learners learn by reducing their cognitive load because it distributes that load among two or more steps. Another kind of method supports learning by eliminating an essential difficulty associated with the learning of interests. We adopt this latter method in our research. First, we specify difficulties associated with learning because we believe that it becomes possible to design how to eliminate or decrease them only after specifying such difficulties.

Recently, several computer-based learning support systems and support methods have been proposed. They help learners master metacognitive activity. It is worth investigating whether or not these systems and methods are designed to eliminate difficulties associated with the learning of metacognitive activity through its clear specification. The concept of metacognitive activity is vague [3, 23]. Several terms are currently used to describe the same basic phenomena (e.g., self-regulation, executive control), or aspects of those phenomena (e.g. meta-memory)[19]. Moreover, these terms are often used interchangeably in the literature [3, 5, 7, 8, 9, 11, 21, 22, 23, 26, 32, 35]. To further complicate matters, two approaches to metacognition exist. On one hand, some researchers consider metacognitive activity as something different from the cognitive activity and attempt to clarify its mechanism [26, 29, 21, 26]. On the other hand, some researchers suppose that metacognitive activity is a similar process to cognitive activity [23, 24]. Such confusion shows that many interpretations of metacognitive activity exist, thereby creating a situation in which difficulties in mastering metacognitive activity are not specified well. Let us take two examples. One example is that the target of a support system changes from the first version and the second version, whereas the authors claim each of them supports mastering metacognitive activity without making the change explicit[11]. The second example is that in spite of the fact that the targets of support are different from one another, the support methods share the same objective. Reciprocal Teaching [4, 27] and ASK to THINK – TEL WHY [21] have the same method and objective: reciprocal tutoring and help of mastering metacognitive activity. By analyzing the learner's cognitive activities when the learner plays a tutor role in both methods, we can see that Reciprocal Teaching causes the learner to observe "the learner's own problem solving process" and ASK THINK to THINK – TEL WHY causes a learner to observe "other learners' problem solving processes." Thus the learner's cognitive activities when playing a tutor role in Reciprocal Teaching and ASK to THINK – TEL WHY vastly differ from each other, even though they both claim to support learning of "metacognition".

Under this situation in which researchers share little common conception of "metacognition", it is difficult to recognize common properties among existing systems to support learning metacognitive activity and their differences. It is almost impossible to reuse one method across systems. Our objective is to support learners in their mastery of metacognitive activity. First, we investigate metacognitive activity itself, which we should support. It fosters our correct understanding of metacognitive activity. Secondly, we specify the factors of difficulties found in mastering it, then discuss functionality for support in eliminating them. We require a framework that represents metacognitive activity from the viewpoint of its difficulty in mastering it. Thereby, we can specify the factors of such difficulties. Using the framework, we can organize existing computer-based support systems and support methods and can understand common and reusable features across systems. However, we do not intend to claim that our framework is valid in terms of cognitive psychology. We provide a common framework for discussing the particularity of each computer-based support system from a technological point of view. Supporting a learner in an attempt to master metacognitive activity would be meaningful if we could gain useful information, based on our framework, for building a computer-based learning support system.

This paper is organized as follows. After analyzing difficulties in performing cognitive activity and distinguishing the factors of that difficulty from others, we construct our framework, which represents metacognitive activity from the viewpoint of the difficulty in mastering it. Finally, we organize existing systems and methods based on that framework.

2. Analyzing Factors of Difficulty in Performing Cognitive Activities

What causes difficulty in performing metacognitive activity? To answer this question, we assume the consideration of metacognitive activity by Lories et al. [24]: 'metacognitive activity processes the contents of (working) memory by standard cognitive process.' In other words, the same architecture can be applicable to both cognitive and metacognitive activities – only their targets are different. The assumption allows us to enumerate factors of difficulties in performing metacognitive activity by analyzing difficulties in performing cognitive activity. In this section, we investigate kinds of cognitive activity, its time-related attributes and its targets, based on Baddeley's Working Memory Model [2]. The term working memory (WM) refers to a system that has evolved for short-term memory and manipulation of information necessary for the performance of such complex tasks as learning, comprehension, and reasoning.

2.1 Kinds of Cognitive Activity and Its Time

Humans have knowledge in the form of so-called operators to achieve goals. Operators are procedures for changing the current state into another that brings us closer to the goal. In general, multiple operators can be applicable to a state, and a critical task is selecting the one to apply. There are some other cognitive activities such as evaluation of the current state which accompany the selection of operators. We divide cognitive activities into five kinds: *rehearsal,* *observation, evaluation, virtual application,* and *selection*. *Observation* is watching something carefully and creating products in WM. It has *focusing* as its subtask. As WM is very rapidly forgotten [2], *Rehearsal* is a critical task for maintaining contents in WM. *Evaluation* is assessing the state of WM and its subtask is *comparison*. By *virtual application,* we mean to apply retrieved operators virtually. *Selection* is choosing appropriate operators among them based on the virtual application results and generating an action-list in WM.

Time attribute is categorized into two measurements of interest to us: one is the time necessary for execution of a cognitive activity; the other is the elapsed time after the execution. Simultaneity of multiple actions and length of an execution time belong to the former; the latter is used to talk about WM and LTM, that is, the memory of action taken is heavily dependent on the elapse time after its execution.

2.2 Targets of Cognitive Activity

Among several cognitive activities, we pay special attention to cognitive operation. By cognitive operation, we mean an operation to generate a new (cognitive) product by applying operators to the contents of WM. For instance, as Fig. 1 shows, the transformation of products-A(t) to products-A(t+1) is a cognitive operation; observation is a cognitive activity. Because a cognitive operation can become the target of (meta)cognitive activity, it is meaningful to distinguish cognitive operations from other cognitive activities. Note here that we do not claim that cognitive operation is not a cognitive activity.

The outside world and a representation stored in WM can be the target of cognitive activity. A representation is encoded information such as a symbolic abstraction of a thing. For instance, observing a car, a person creates a representation of it in WM. A representation is categorized into three according to its target: *outside world*, *cognitive operation*, and *LTM*. By *outside world,* we mean a representation which is created by observing the outside world. A person creates a representation whose target is *cognitive operation* if one observes how another person solves a problem. Concerning *cognitive operation*, it is also important to distinguish one that is created by observing one's own cognitive operation from one that is created by observing others' cognitive operation.

There are two kinds of targets of cognitive activity: an object and a process. An object includes an abstract one. Both objects and processes have two subcategories respectively: outside-world objects and inner-world (mental) objects; outside-world processes and inner-world (mental) processes.

As the number of factors associated with an activity increases, a learner performs the activity with increasing difficulty.

3. Constructing a Framework of Difficulty in Performing Metacognitive Activity

We construct our framework representing the difficulty in performing metacognitive activity by combining all kinds of cognitive activities with their targets, though it is not perfect. First, we will describe our two-layer model of cognitive activity (Fig. 1).

3.1 A Two-Layer Model of Cognitive Activity

This subsection presents a model of problem solving based on Baddeley's Working Memory Model [2]. An individual initially observes a task condition and creates elements in WM (products-A(t)) as its model when a learner solves a problem. That learner evaluates the problem and investigates if he has some domain knowledge useful to accomplish the task, and if the result is positive, then he retrieves applicable operators from his knowledge base and

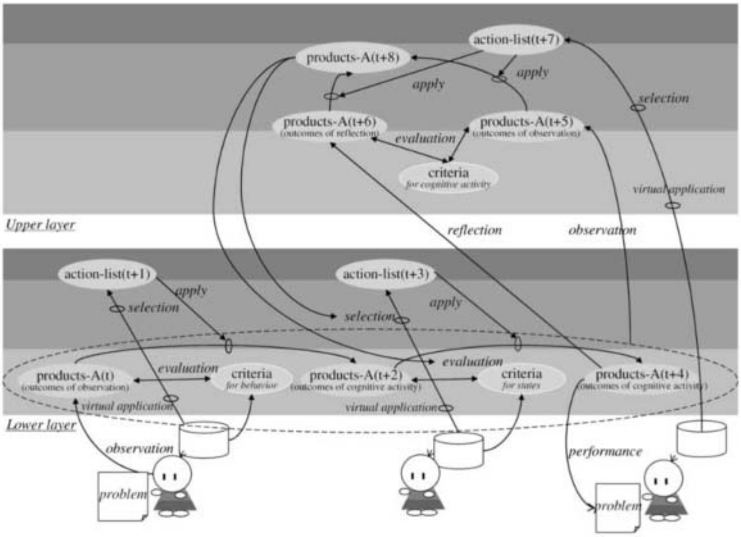


Figure 1. A Computational Model of Problem Solving

applies them to the WM elements (products-A(t)) virtually. Next, evaluating the application results, the learner selects appropriate operators and makes an action-list (action-list(t+1)). Then, the learner applies operators in the action-list to the elements, and creates new elements in his own WM (products-A(t+2)). The process is repeated until achieving the goal, and finally he engages in an observable behaviour (concrete actions to the real world) based on the final elements (products-A(t+4)). Thus, normal problem solving performs cognitive activities that update the state in the lower layer of WM.

Concerning the lower layer of Fig. 1, elements in WM are usually not the target of "observation" but the target of other cognitive activities. However, we occasionally need to observe elements in WM consciously. In this case, where do we create the results of observation? Generally, observation means to observe something in an outside world and to create corresponding elements in WM. As described in the above, the target of observation and its products belong to different worlds from each other. Accordingly, observation of elements in WM creates elements in another world. For that reason, it would be valid to suppose a two-layer model of WM [18]. Supposing a two-layer model, when one observes elements at the lower layer, he can create new elements at the upper layer in WM (products-A(t+5)). Such observation is sometimes called reflection. Many definitions in the literature of reflection exist. Most concur that it is an active, conscious process. Schon divides reflection into two kinds: reflection-in-action (thinking on your feet) and reflection-on-action (retrospective thinking) [29]. We also divide observation of elements in WM into activities of observation and reflection. The former is called conscious observation, to observe a body of existing elements in WM and their operation process and create elements at the upper layer (Products-A(t+5)). We call the latter reflection. By reflection, we mean retrospective creation of elements at the upper layer. One observes some existing elements at the lower layer. Based on them, one infers a past cognitive operation process and creates elements at the upper layer (Products-A(t+6)). For instance, if we are shown some mistakes, we occasionally call and review retrospectively problem-solving processes to identify the reason for the failure.

3.2 Our Framework for Metacognitive Activity

Table 1 shows factors of difficulty in performing a cognitive activity based on our framework that comprises two dimensions such as cognitive activities and their targets. Whatever the

Table 1. Difficulty in Performing a Cognitive Activity

Target	Cognitive Activity	Rehearsal	Observation	Evaluation	Virtual Application	Selection
Out side World	Ordinary Object					
	Resulting object of others' cognitive operations		(d2)			
	Ordinary process		(d1)			
	Others' cognitive operation processes		(d1) (d2)			
Repre sen tation	LTM object (a representation of the retrieved thing)		(d4)			
	Representation of ordinary object at an outside world	(d3)		(d4)	(d4)	(d4)
	Representation of ordinary process at an outside world	(d3)		(d4)	(d4)	(d4)
	Representation of resulting object of others' cognitive operations	(d3)		(d4) (d5)	(d4) (d5)	(d4) (d5)
	Representation of others' cognitive process	(d3)		(d4) (d5)	(d4) (d5)	(d4) (d5)
	Resulting object of one's own cognitive operation	(d3)	(d4) (d6)			
	Representation of resulting object of one's own cognitive operation	(d3)		(d4)(d6)(d7)	(d4)(d6)(d8)	(d4) (d6)
	One's own cognitive operation process	(d3)	(d4) (d6)			
	Representation of one's own cognitive operation process	(d3)		(d4)(d6)(d7)	(d4)(d6)(d8)	(d4) (d6)

d1: Segmentation of process
d2: Invisibility
d3: Simultaneous processing with other activities
d4: Simultaneous processing with rehearsal
d5: Cognitive operation (inference)
d6: A two-layer WM
d7: Acquisition of criteria for cognitive activity
d8: Influence on virtual application at a lower layer

targets are, difficulties exist in performing cognitive activities to some extent. First, we illustrate the relative difficulties in performing cognitive activities. Because *selection* is performed simultaneously with *rehearsal*, it is more difficult than either *observation* or *evaluation*. Because *virtual application* is performed with *rehearsal* simultaneously and is repeated until finding appropriate operators, it is more difficult than *selection*.

Table 1 shows that targets of a cognitive activity are classifiable into two types: the outside world and representation. A cognitive activity that targets the outside world includes only *observation*. Generally speaking, observing a process is more difficult than observing an object because extraction of a process is essentially more difficult than extraction of an object (d1). For instance, a motor skill such as typing using a keyboard is a pre-packaged sequence of actions. It is difficult to extract a part of the action from it. In any case, *observation* of the outside world is the easiest.

By observing LTM and retrieving information such as operators, a representation of what is retrieved is created in WM. Observing LTM is more difficult than observing the outside world because it requires synchronous processing with *rehearsal* (d4) of elements in WM. The other four cognitive activities for a representation, which is created by observing LTM, present similar difficulty as cognitive activities for a representation, which is created by observing the outside world. For that reason, we omit them from Table 1.

By observing ordinary objects or processes of the outside world, corresponding representations are created: "representation of an ordinary object of the outside world" and "representation of an ordinary process of the outside world." They can become the target of *rehearsal, evaluation, virtual application,* and *selection*. Regarding them, one factor of difficulty in performing each kind of activities is synchronous processing (d4).

Because other persons' cognitive operations are invisible (d2), representations concerning others' cognitive operations, which a learner creates in WM by observing others' cognitive operations, becomes incomplete. The learner may supplement them with her inference. Therefore, cognitive activities for representations concerning others' cognitive operations are more difficult than representations of ordinary objects (or processes) of the outside world. Whatever the targets are, the difficulty encountered in performing *rehearsal* is the same. As Table 1 shows, factors of difficulty in performing cognitive activity for representations concerning others' cognitive operations include the synchronous processing of rehearsal (d4) and cognitive operations (inference) (d5).

As described in 3.1, when one observes "one's own cognitive operation", the inner world in which ordinary cognitive activities are executed becomes another outside world, and WM is made to be two-layered. Considering the two-layer WM, synchronous processing is essential, that is, one performs cognitive activity for elements at the upper-layer while maintaining elements at a lower-layer. As the factors of difficulty in performing cognitive activities for "one's own cognitive operation", we identify synchronous processing (d4) and a two-layered WM (d6). In addition, *evaluation* has a special factor of difficulty: acquisition of "criteria for cognitive activity (d7)" at the upper layer. A learner can learn by evaluating elements at the lower layer and by solving common problems that are often encountered if a learner needs to acquire only "criteria for states" at the lower layer. However, such is not the case. A learner needs to learn "criteria for cognitive activity" as well and he rarely has a chance to learn "criteria for cognitive activity", which causes acquisition of "criteria for cognitive activity (d7)" to become more difficult. *Virtual application* of an operator at the upper layer affects virtual application of an operator at the lower layer (d8), which must be resolved during processing at the upper layer. Therefore, virtual application at the upper layer is the most difficult in cognitive activities. It causes *selection* of an appropriate operator to become difficult (d9) at the upper layer.

4. Organization of Existing Computer-based Support Systems and Methods Based on Our Framework

Using our framework, we analyze existing computer-based support systems and methods to clarify the correspondence between them and factors of difficulty and to specify their targets. According to the correspondence, we categorize factors of difficulty into two from a unified viewpoint: those which some support systems already intend to eliminate and those which no systems intend to eliminate. The categorization can reveal what factors remain without support.

Although we have analyzed some representative support systems: MIRA [11], Algebraland Computer System [5, 7], Geometry Tutor [1], Interactive History [14], Intelligent Novice Tutor [25], and Error Based Simulation [13], we describe only three examples because of space limitations in this paper. ASK to THINK – TEL WHY is an inquiry-based tutoring model. A tutor guides learners by asking a question using a given template of five kinds of questions. Tutees only answer questions. King claims that tutees become aware of metacognitive activity in answering self-regulation questions (SR-Qs) [21]. Asking SR-Qs is training also for the tutor to observe *others' cognitive operation process* because he must determine the timing of asking an SR-Q. Ideally, a tutor should observe *his own cognitive operation process*, but factors of difficulty exist (d4 and d6 in Table 1). The target of observation is shifted from *one's own cognitive operation process* to *others' cognitive operation processes* to eliminate these factors. The tutor's SR-Qs induce tutees to observe *their own cognitive operation processes*. The other four questions by a tutor allow tutees to observe *resulting objects of one's own cognitive operations*. Tutors' questions reduce tutees' cognitive loads of cognitive activity at the upper layer. Tutees' answers of these questions also allow the tutor to evaluate tutees' results of cognitive operation. It also means to shift the target of evaluation from one's own cognitive operation to others for eliminating difficulty.

In Reciprocal Teaching [4, 27], learners in a small group take turns playing the discussion leader role and a monitoring role for the goal of understanding a text. For a discussion leader role, a learner externalizes his comprehension, such as in a summary. It is training for observing or evaluating results of his own cognitive operation; it incurs a heavy cognitive load. For that reason, the method allows a teacher to advise a discussion leader if

Table 2. Correspondence between Difficulties and Supports

Target	Cognitive Activity	Existing support methods and support systems	
		Support that reduces difficulty	Other support
Resulting object of one's own cognitive operation	Observation		ASK-the other-Q (tutees) Reciprocal-summary (leader)
Representation of resulting object of one's own cognitive operation	Evaluation	ASK-tutees' answers of the other-Qs (tutor)	Reciprocal-evaluation (leader)
	Virtual Application		
	Selection		
One's own cognitive process	Observation	[Elimination of d4,d6 by shifting a target] ASK-SR-Q (tutor) Reciprocal (monitors) Our method (monitors) [Elimination of d4,d6 by externalizing] Kitchen-third (learners) Algebraland (Search Space Window) Geometry Tutor IH (navigation window) Our method (externalization tool)	ASK -SR-Q (tutees) Reciprocal-others' advice (leader) Kitchen-third (teacher) Kitchen-fourth (learners) Geometry Tutor (complete proof) MIRA (pre&post reflection) IH (notation) EBS (simulation)
Representation of one's own cognitive process	Evaluation	[Elimination of d4,d6 by shifting a target] ASK-SR-Q-evaluation (tutor) Reciprocal-advice (others) Our method (monitors) [Elimination of d7] Our method-a template of Q (monitors)	Reciprocal-others'advice (leader) Kitchen-fourth (learners) Our method-monitors' Q (solver)
	Virtual Application		
	Selection		

necessary. The method also provides an opportunity for other learners to observe and evaluate the discussion leader's summary. It is training that monitors observe and evaluate *others' cognitive operation process*. However Palincsar et al. seems not to have understood such an effect, that is, Reciprocal Teaching eliminates factors of difficulty (d4, d6) by shifting the target from *one's own cognitive operation* to *others' cognitive operations*, exactly as observed in ASK to THINK – TEL WHY.

Shoenfeld describes the "Kitchen Sink" approach as "four classroom techniques that focus on metacognition" [30]. The "Kitchen Sink" approach reduces a learner's cognitive load by dividing a learning process into two or more sequential stages. The first and second techniques show the problem solving process of a novice and an expert. They pull the trigger at an awareness of metacognitive activity and get learners motivated to master it. The third technique is a practical demonstration of metacognitive activity by an expert and externalizes the *learners' own cognitive operation processes*. The fourth technique gives an opportunity to perform metacognitive activity by asking a question. In summary, Kitchen Sink does not try to eliminate difficulties associated with learning of metacognitive activity.

Through these analyses, we clarify the correspondence between the difficulty in performing metacognitive activity and existing support systems and methods. We find that most of the support systems and methods help eliminate factors of difficulty (d4, d6). In addition, some of these reduce the difficulty of evaluating *one's own cognitive operation* by shifting the target from it to *others' cognitive operation*. Nevertheless, no systems and methods exist that help a learner acquire the criteria for cognitive activity (d7) and master virtual application (d8) and selection of appropriate operators (d9) at the upper layer of WM.

We have designed our support method by eliminating the difficulties (d4, d6 and d7) including the adoption of those effective ways in the existing support systems and methods with explicit explanation of what difficulty we are going to eliminate and how to realize it. Furthermore, our method has been designed to gradually increase individual cognitive load [15, 17, 18, 19].

5. Conclusion

We have tried to uncover the correspondence between existing systems for supporting learning of metacognitive activity and factors of its difficulty based on the framework we have developed. The correspondence indicates that existing support methods and systems address different targets with the same goal of helping learners acquire metacognitive activity. Our framework can also contribute to a shared understanding of research on assisted learning of metacognitive activity and accumulation of the research results.

References

[1] Anderson, J. R., Boyle, C. F., Farrell, R., & Reiser, B. J. (1987). Cognitive principles in the design of computer tutors. In P. Morris (ed.), Modelling Cognition, Wiley.
[2] Baddeley, A. (1986). Working memory. Oxford: Clarendon Press.
[3] Brown, A. (1987). Metacognition, executive control, self-regulation, and other more mysterious mechanisms. In Weinert, F.E. & Kluwe, R. H. (eds.) Metacognition, motivation, and understanding. (pp.65-116). NJ: LEA.
[4] Brown, A. L. & Palincsar, A. S. (1989) Guided, cooperative learning and individual knowledge acquisition. In knowing, learning, and instruction: essays in honor of Robert Glaser, LEA
[5] Brown, J. S.(1985). Process versus product: a perspective on tools for communal and informal electronic learnig, J. Educational computing research, Vol.1(2).
[6] Carver, C. S. & Scheier, M. F. (1998). On the self-regulation of behavior. New York: Cambridge Univ. Press.
[7] Collins, A., Brown, J. S. (1988). The computer as a tool for learning through reflection. In Mandl, H. Lesgold, A. (eds.) Learning issues for intelligent tutoring systems.Springer-Verlag.
[8] Davidson. J. E., Deuser, R. & Sternberg, R. J. (1994). The role of metacognition in problem solving. In Metcalfe & Shimamura (Eds.) Metacognition. Cambridge: MIT Press. 207-226.
[9] Flavell, J. H. (1976) Metacognitive aspects of problem-solving. In Resnick, L. B. (ed.), The nature of intelligence. NJ: LEA. 231-235.
[10] Flavell, J. H. (1987). Speculations about the nature and development of metacognition. In Weinert, F. E. and Kluwe, R. H. (Eds.), Metacognition, motivation, and understanding. NJ: LEA. 21-29.
[11] Gama, C. (2004). Metacognition in interactive learning environments:the reflection assistant model. Proc. of ITS2004.
[12] Hacher, D. J. (1998). Definitions and Empirical Fpundations. In Hacker, D. G., Dunlosky, J. and Graesser, A. C. (Eds.) Metacogniton in educational theory and practice. NJ:LEA. 1-23.
[13] Hirashima T., Horiguchi T. (2003) Difference visualization to pull the trigger of reflection. Proc. of AIED2003.
[14] Kashihara A., Hasegawa S. (2004). Meta-learning on the web. Proc. of ICCE2004.
[15] Kayashima, M., Inaba, A. (2003). How computers help a learner to master self-regulation skill? Proc. of CSCL2003.
[16] Kayashima, M., Inaba, A. (2003). Difficulties in mastering self-regulation skill and supporting methodologies. Proc. of AIED2003.
[17] Kayashima, M., Inaba, A. (2003). Towards helping learners master self-regulation skills. supplementary Proc. of AIED2003.
[18] Kayashima, M., Inaba, A. (2003). The model of metacognitive skill and how to facilitate development of the skill. Proc. of ICCE2003.
[19] Kayashima, M., Inaba, A. and Mizoguchi, R.(2004). What is metacognitive skill? – Collaborative learning strategy to facilitate development of metacognitive skill. Proc. of ED-MEDIA2004.
[20] Kayashima, M., Inaba, A. and Mizoguchi, R. (2004). Towards shared understanding of metacognitive skill and facilitating its development. Proc. of ITS2004.
[21] King, A. (1998). Transactive peer tutoring: distributing cognition and metacognition, J. of Educational Psychology Review, 10(1).
[22] Kluwe, R. H. (1982). Cognitive knowledge and executive control: metacognition. In Griffin, D. R. Animal Mind – Human Mind (ed.) New York: Springer-Verlag. 201-224.
[23] Livingston, J. A. (1997). Metacognition http://www.gse.buffalo.edu/fas/shuell/cep564/Metacog.htm.
[24] Lories, G., Dardenne B., and Yzerbyt, V. Y. (1998). From social cognition to metacognition. In Yzerbyt, V. Y., Lories, G., Dardenne, B. (eds.) Metacognition, SAGE Publications Ltd.
[25] Mathan, A. and Koedinger, K. (2003) Recasting the feedback debate: benefits of tutoring error detection and correction skills, Proc. of AIED2003.
[26] Nelson, T. O., Narens, L. (1994). Why investigate metacognition? In Metcalfe, J. and Shimamura, A.P. Metacognition, (eds.) (pp.1-25). MIT Press.
[27] Palincsar, A. S. and Herrenkohl, L. R. (1999). Designing collaborative contexts: lessons from three research programs. In O'Donnell, A. M. and King, A. (eds.) Cognitive Perspectives on Peer Learning, Mahwah, NJ:LEA.
[28] Rivers, W. (2001). Autonomy at all costs: an ethnography of metacognitive self-assessment and self-management among experienced language learners. Modern Language Journal 85(2), 279-290.
[29] Schon, D. A. (1983). The reflective practitioner. Basic Books, Inc.
[30] Schoenfeld, A. H. (1987). What's all the fuss about Metacognition. In A.H. Shoenfeld (ed.) Cognitive science and mathematics education, Lawrence Erlbaum Associates.
[31] Schraw, G. (1998). Promoting general metacognitive awareness. Instructional Science. 26(2), 113-125.
[32] Van Zile-Tamsen, C. M. (1994). The role of motivation in metacognitive self-regulation. Unpublished manuscript, State University of New York at Buffalo.
[33] Van Zile-Tamsen, C. M. (1996). Metacognitive self-regualtion and the daily academic activities of college students. Unpublished doctoral dissertation, State University of New York at Buffalo.
[34] Winne, P. H. and Hadwin A. F. (1998). Studying as self-regulated learning. In Hacker, D. J., Dunlosky, J. and Graesser, A. C. (eds.) Metacognition in educational theory and practice. NJ: LEA. 277-304.
[35] Yzerbyt, V. Y., Lories, G. and Dardenne, B. (eds.) (1998). Metacognition, SAGE Publications Ltd.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Matching and Mismatching Learning Characteristics with Multiple Intelligence Based Content

Declan KELLY [1], Brendan TANGNEY [2]
[1] National College of Ireland, Dublin, Ireland, [2] University of Dublin, Trinity College, Ireland
[1]dkelly@ncirl.ie, [2]tangney@tcd.ie

Abstract: Research informs us that learning characteristic differ, that knowledge is processed and represented in different ways and that students prefer to use different types of resources in distinct ways. However, building Adaptive Educational systems that adapt to different learning characteristics is not easy. Major research questions exist such as: how are the relevant learning characteristics identified, how does modelling of the learner take place and in what way should the learning environment change for users with different learning characteristics?

EDUCE is one such system that addresses these challenges by using Gardner's theory of Multiple Intelligences (MI) as the basis for dynamically modelling learning characteristics and for designing instructional material. This paper describes a research study, using EDUCE, that explores the effect of using different adaptive presentation strategies and the impact on learning performance when material is matched and mismatched with learning preferences. The results suggest that students with low levels of learning activity, and who use only a limited number of the resources available, have the most to benefit from adaptive presentation strategies and that surprisingly learning gain increases when they are provided with resources not normally preferred.

1 Introduction

Educational research tells us "one size does not fit all" [15]. It informs us that learning characteristics differ, that knowledge is processed and represented in different ways, and that learners use different types of resources in distinct ways [16]. Research also suggests that it is possible to diagnose a student's learning style and that some students learn more effectively when instruction is adapted to the way they learn [14].

Within the field of technology enhanced learning, adaptive educational systems offer an advanced form of learning environment that attempts to meet the need of different students. Such systems build a model of the student's knowledge, goals and preferences, and use the generated model to dynamically adapting the learning environment for each student in a manner that best supports learning [1]. Several adaptive educational systems that adapt to different learning characteristics have been developed [5][18][11]. However building such systems is not easy and major research questions include: how are the relevant learning characteristics identified, how modelling of the learner take place and in what way shall the learning environment change for users with different learning characteristics [12].

EDUCE [6] is an adaptive intelligent educational system that addresses these challenges by using Gardner's theory of Multiple Intelligences (MI) as the basis for dynamically modelling learning characteristics and for designing instructional material [4]. The theory of Multiple Intelligences reflects an effort to rethink the theory of measurable intelligence embodied in intelligence testing. It supports suitably the motivation behind EDUCE: that intelligence is not a fixed static entity, but something that resides inside a person, and can be enhanced significantly through education and awareness. It is also a rich concept that offers a framework and a language for developing adaptive educational systems that supports creative, multimodal teaching [10]. In the past 20 years since its inception, its use in the classroom has been significant [2] but, surprisingly, its application to online learning and adaptive educational systems is still in the early stages of research [6].

This paper describes the results of an empirical study that explores the following research questions.

- What is the effect of using different adaptive presentation strategies rather than giving the learner complete control over the learning environment?
- What is the impact on learning performance when resources are matched and mismatched with learning preferences?

In particular, the study examines the relationship between the adaptive presentation strategy, the choice of resources available and the learning performance of science school students aged 12 to 14 using different versions of EDUCE. The adaptive presentation strategy involves matching and mismatching students with resources they prefer and do not prefer to use. The level of choice determines the number of resources a student has access to and the manner in which EDUCE adaptively guides the student to view a particular resource first. Learning performance is defined by the learning gain and learning activity. Learning gain is measured by a pre and post-test and learning activity is determined by the navigation profile and the number of available resources used.

The results suggest that teaching strategies that encourage students to use a broad range of resources are the most effective. In particular, they suggest that students with low levels of learning activity have the most to benefit from adaptive presentation strategies and that surprisingly learning gain increases when they are provided with resources not normally preferred.

2 EDUCE

In EDUCE, a student model of learning characteristics is created using the MI theory. The theory identifies eight intelligences that are involved in solving problems, in producing material such as compositions, music or poetry and other educational activities. In contrast to learning styles, intelligences refer to abilities in what one can do such as execute skills or strategies, whereas styles refer to preferences in the use of abilities. Moreover, an intelligence is usually limited to a particular domain of content, such as verbal ability, whereas style cuts across domains of ability. Currently EDUCE uses the four intelligences in modelling the student:

- Logical/Mathematical intelligence (LM) - This consists of the ability to detect patterns, reason deductively and think logically.
- Verbal/Linguistic intelligence (VL) - This involves having a mastery of the language and includes the ability to manipulate language to express oneself.
- Visual/Spatial intelligence (VS) - This is the ability to manipulate and create mental images in order to solve problems.
- Musical/Rhythmic intelligence (MR) - This encompasses the capability to recognise and compose musical pitches, tones and rhythms.

The three intelligences, LM, VL and VS were chosen as they reflect the abilities that are historically designated as intelligences. The musical/rhythmic intelligence was chosen because it is not considered as an intelligence that can be used to deliver and inform the design of content yet the emotive power of music is widely acknowledged [3].

The static MI profile of each student is determined by getting the student to first complete, before starting the tutorial, the MIDAS MI inventory [17]. EDUCE also builds a dynamic model of the student's MI profile by observing, analysing and recording the student's choice of MI differentiated material. Other information also stored in the student model includes the navigation history, the time spent on each learning unit, answers to interactive questions and feedback given by the student on navigation choices.

EDUCE holds a number of tutorials designed with help of subject matter experts. Each tutorial contains a set of content explaining a particular subject area. For the experiment described in this paper, Science is the subject matter. A tutorial consists of learning units that explain a particular concept. In each unit there are four different sets of learning resources, each based predominantly on one of the intelligences. The different resources explain a topic from a different angle or display the same information in a different way. Different instructional design strategies and techniques were used to create the content [7]. For verbal/linguistic content it was the use of explanations, descriptions, highlighted keywords, term definitions and audio recordings. For logical/mathematical content it was the use of number, pattern recognition, relationships, questioning and exploration. For visual/spatial content it was the use of photographs, pictures, visual organisers and colour. For musical/rhythmic content it was the use of musical metaphors, raps and rhythms. All resources developed were validated and identified as compatible with the principles of MI theory by expert practitioners.

Each learning unit consists of several distinct stages. The first stage aims to attract the learner's attention, the second stage provides a set of different MI resources, the third stage re-enforces the key message in the lesson and the final stage presents interactive questions on the topic. After accessing the second stage, students may repeatedly go back and use the same or different MI resource. The presentation strategy controls the movement from the first to the second stage. Different strategies guide students to resources they like to use and do not like to use. In this process, different versions of EDUCE can be used. One version of EDUCE uses the static MI profile to identify the learning preference, another version uses the dynamically generated student model. The dynamic student model is generated from a set of navigational and temporal features that act as behavioural indicators of the student's learning characteristics. EDUCE's predictive engine [8], with these features as input and the Naïve Bayes algorithm as its inference engine, dynamically detects patterns in the learning behaviour and determines the learner's preferences.

3 Experimental Design

The experiment was designed in such a manner to explore the effect of different adaptive presentation strategies and to determine the impact on learning performance when resources were matched with preferences. In particular it was set up to explore the impact of the two independent variables, presentation strategy and level of choice, on the dependent variable, learning performance. Different configurations of EDUCE were used to support the different values of the independent variables. The effect of other variables such as MI Profile and prior ability on learning performance was also examined.

The presentation strategy for delivery material encompasses two main strategies.

1. *Most preferred*: - showing resources the student prefers to use
2. *Least preferred*: - showing resources the student least prefers to use

For each learning unit, there are four MI based learning resources. The MI profile and the presentation strategy determine which resource is shown first.

The second independent variable is the level of choice. There are three different levels of choice provided to different groups corresponding to the different adaptive versions of EDUCE:

1. *Single* – student is only able to view one resource. This is adaptively determined by EDUCE based on an analysis of the static MI profile.
2. *Inventory* - student is first given one resource but has the option to go back and view alternative resources. The resource first given to the student is determined by EDUCE based on the analysis of the MI inventory completed by the student. The *Inventory* choice level is the same as the *Single* choice level but with the option of going back and viewing alternative resources.
3. *Dynamic* – the student is first given one resource but has the option to go back and view alternative resources. The resource first given to the student is determined by using the dynamic MI profile that is continuously updated based on the student's behaviour. The predictive engine within EDUCE identifies the most preferred and least preferred resource from the online student computer interaction.

Learning performance is defined by the learning gain and learning activity. To calculate the relative learning gain each student before and after a tutorial sits a pre-test and post test. The test for the pre-test and post-test is the same and consists of questions that appear during the tutorial. Learning activity is determined by the navigation profile. It is a measure of the different panels visited, the number of different resources used, the reuse of particular resources and the direction of navigation. The questions are multi-choice question with four options. Learning activity is analysed to provide informed explanations on learning gain. Table 1 displays the variables used in the study and their values.

The experiment was conducted over three days. On Day-1, students completed the MIDAS MI Inventory. On Day-2, each student spent on average 22 minutes, with no significant difference betweent the different groups, exploring one tutorial. The session was preceded by a pre-test and followed by a post-test. The pre-test and post-test had the same 10 multi-choice questions, which were mostly factual. Day-3 repeated the same format as Day-2, except that the student explored a different tutorial. On different days, the most preferred and least preferred presentation strategies were used. Students were randomly assigned to one of the three groups defined by the levels of choice. To ensure order effects are balanced out, students are also assigned to systematically varying sequence of conditions. The design of the experiment can be described as a mixed between/within subject design with counterbalance.

Variable	Value
Presentation Strategy	Least Preferred, Most Preferred
Choice Level	Single, Inventory, Dynamic
Relative Learning Gain	(Post test score-pre test score)/pre test score
Activity Level	% of resources used
Activity Groups	Low, Medium and High Activity
Prior Ability	Score from previous class test
Dominant Intelligence	Highest ranking intelligence as recorded by MIDAS Inventory

Table 1: Variables used and their values

5 Results

47 boys from one mixed ability school participated in the study. The average age was 13 and the study was conducted as part of normal class time and integrated into the daily school curriculum. 20 used the single choice version, 18 the inventory choice version and 9 the dynamic choice version. The results were analysed from two perspectives:

- The effect of presentation strategy and level of choice on learning gain
- The relationship of learning activity and gain

5.1 Effect of choice and presentation strategy on learning gain

The results were first analysed to determine the effect of different adaptive strategies on learning performance. It was expected that students would have greater learning gain when guided to resources they prefer instead of those they do not prefer. It was also expected that the groups (inventory and dynamic) with access to a range of resources would have higher learning gain than the group (single) who did not. Furthermore, it was also expected that the group (dynamic) who were guided to resources based on a dynamic model of behaviour would have higher learning gain than all other groups.

To explore the effects of the two independent variables, choice and presentation strategy, a mixed between-within ANOVA was conducted. The relative gain score obtained under the two presentation strategies, least and most preferred, were compared.

With the relative gain scores, there was a significant within subject main effect for presentation strategy: Wilks Lambda: 0.897, F = 4.944 (1, 43), p = .031, multivariate eta square = .103. The mean relative gain score at the least preferred sitting (M=76.2, SD=99.5) was significantly greater than the score at the most preferred sitting (M=38.9, SD=51.9). The eta square suggests a moderate to large effect size. Figure 1 plots the relative gain for the least and most preferred strategies. It shows that for all groups, and in particular for the inventory and dynamic choice groups, that the relative gain is greater in the least preferred condition. The differences between the different choice groups were not significant.

Surprisingly, the results indicate that students learn more when first presented with their least preferred material rather than their most preferred material, in contradiction to the original hypothesis.

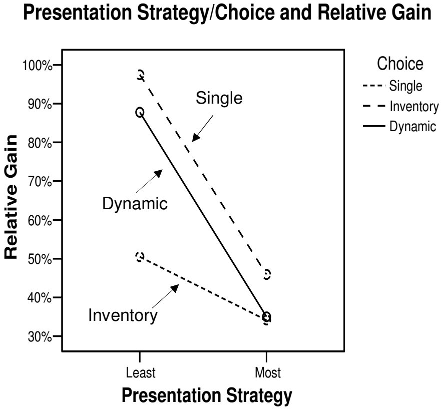


Figure 1: Plot of Relative Gain for least/most presentation strategy

5.2 *Learning activity and performance*

To investigate the reasons for the difference in learning gain with the least/most preferred presentation strategies, learning activity was analysed. The purpose was to explore if students using a large variety of resources had the same learning gain as students who used only the minimum. It was expected that the activity level would increase with the least preferred presentation strategy, and that higher learning activity would result in increased learning gain for all students.

To determine the overall activity level, the average of the percentage of resources used in the least and most condition is calculated. Three categories are defined for activity: low, medium and high. The cut points for each category were determined by dividing students into three equal groups based on their activity level. Typically, a student in the low activity group would look at only one resource per learning unit, a student in the high activity group would on average look at two resources per unit and in a student in the medium activity group would be somewhere in between. Only the inventory and dynamic choice groups were included in the analysis as it is irrelevant to calculate the activity level for the single choice group, having access to only one resource.

A two way mixed between-within ANOVA was conducted to explore the effect of activity level and presentation strategy on relative gain. The means and standard deviations of the relative gain scores are presented in Table 2. There was a significant within subject main effect for presentation strategy: Wilks Lambda: 0.818, F = 5.332 (1, 24), p = .03, multivariate eta square = .182. There was also a within-subject interaction effect between relative gain score and activity level, however it was only significant at the p<.1 level: Wilks Lambda: 0.808, F = 2.851 (2, 24), p = .077. This interaction effect was primarily due to the fact that low activity learners had a higher relative gain at the least preferred sitting than at the most preferred sitting. For medium and high activity learners, despite the learning gain been slightly higher at the least preferred sitting, the presentation strategy had no statistically significant impact on learning gain.

Figure 2 plots the relative gain for the different activity groups in the least and most preferred condition. Its shows how students with low activity have higher relative learning gain when given least preferred resources first. Students with medium and high activity have the same relative gain in both the least and most preferred conditions. The results indicate that students with low learning activity levels benefit most when they are encouraged to use resources not normally used.

Analysis was also conducted to determine if presentation strategy had an impact on learning activity for the different activity groups. Figure 3 shows how activity levels remain similar in both the least and most preferred presentation conditions. This was supported by a correlation between the activity levels in both conditions (r=.65, p<.01). It suggests the presentation strategy did not influence learning activity and that the difference in learning gain for low activity learners may be dependent on the type and variety of resource provided.

Together, the results indicate that the presentation strategy had a different effect for students with different levels of activity. Students with high and medium activity levels were not influenced by presentation strategy. In contrast, the presentation strategy had a significant impact on low activity students, who had larger increases in learning gain when encouraged to use resources not normally preferred. The implications are that students with low levels of learning activity have the most to benefit from adaptive presentation strategies.

	Least Relative Gain		Most Relative Gain		
Activity	Mean	Std. Dev.	Mean	Std. Dev.	N
Low	174.07	160.75	46.30	48.42	9
Medium	48.07	69.05	30.00	40.72	9
High	60.56	49.64	50.56	73.59	9
Total	94.23	116.25	42.28	54.58	27

Table 2 Relative gain for different activity groups

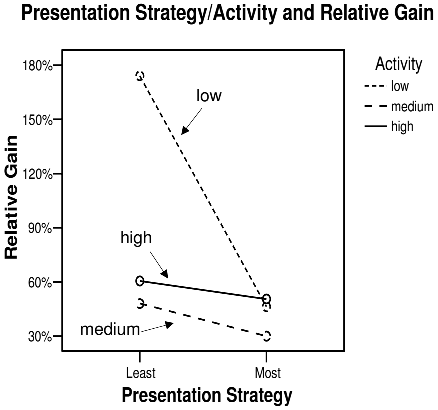


Figure 2. Relative gain for different groups in least/most preferred conditions

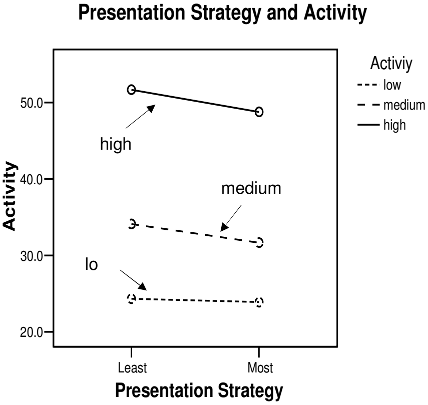


Figure 3. Activity and least/most presentation strategy for different activity groups

6 Discussion

The experiment was conducted to explore the effect of presentation strategy and level of choice on learning performance. Nothing conclusive could be said about the effect of level of choice as the results were not statistically significant. However, when exploring the impact of presentation strategy, the relative gain scores in the least and most preferred conditions were significantly different. Unexpectedly, the results suggest that students learn more in the least preferred condition rather than in the most preferred condition.

To further analyse this surprising result, students were divided into groups defined by their learning activity or the number of resources they used during the tutorial. On exploring the relative gain for different activity groups in the least and most preferred condition, further insight was revealed. It was only students with low activity levels who demonstrated different relative learning gains, with significantly greater learning gain in the least preferred condition. The result suggests that students with low levels of learning activity can improve their performance when adaptive presentation strategies are in use.

A further analysis was conducted to determine if presentation strategy had an impact on learning activity. For the different activity groups, there was no significant difference in the levels of activity in the least and most preferred conditions. The result indicates that presentation strategy may not influence learning activity, and that low activity learners will remain low activity learners regardless of the resource they use, least preferred or most preferred. Combining this with the fact that the relative learning gain is higher in the least preferred condition, it suggests that the type of resource used may make a difference.

Taken together, the results suggest that using adaptive presentation strategies to provide students with a variety of resources that are not preferred enhances the performance of low activity learners. This, somewhat, surprising result is in contrast to the traditional MI approach of teaching to strengths and suggests that the best instructional strategy is to provide a variety of resources that challenge the learner. However this may not be as surprising when one considers the motivational aspects of games and their characteristic features. Challenge is one of the key motivational characteristics of games [13] and it maybe that in education too, challenge at the appropriate level is also needed.

7 Conclusion

This paper described an experimental study that explored the impact of presentation strategy and different kinds of adaptivity on learning performance. The results suggest that students with low levels of learning activity can improve their performance when adaptive presentation strategies are in use. They suggest that challenging students may be a key aspect of learning environments.

Future work will involve exploring further the role of challenge in learning environments. It will involve determining the influence of different types of resources on individual learners and their effect on learning performance. More research will also be conducted to explore what influences learning activity, and to determine if strategies that increase learning activity also increase learning gain.

Acknowledgements: Many thanks to the teachers and students of St. Benildus College, Stillorgan, Dublin, Ireland.

References

[1] Brusilovsky, P. (2001): Adaptive Hypermedia. User Modeling and User-Adapted Instruction, Volume 11, Nos 1-2.

[2] Cambpell, L. & Campbell, B. (2000): Multiple Intelligences and student achievement: Success stories from six schools, Association for Supervision and Curriculum Development.

[3] Carroll, K. (1999). Sing a Song of Science. Zephyr Press.

[4] Gardner H. (2000): Intelligence Reframed: Multiple Intelligences for the 21st Century. Basic Books

[5] Gilbert, J. E. & Han, C. Y. (1999): Arthur: Adapting Instruction to Accommodate Learning Style. In: Proceedings of WebNet'99, World Conference of the WWW and Internet, Honolulu, HI.

[6] Kelly, D. & Tangney, B. (2002): Incorporating Learning Characteristics into an Intelligent Tutor. In: Proceedings of the Sixth International on ITSs, ITS2002 p729-738

[7] Kelly, D. & Tangney, B. (2003): A Framework for using Multiple Intelligences in an ITS. In: Proceedings of EDMedia'03 p2423-2430

[8] Kelly, D. & Tangney, B. (2004): Predicting Learning Characteristics in a Multiple Intelligence based Tutoring System. In: Proceedings of the Seventh International on ITSs, ITS2004 p679-688

[9] Kelly, D. & Tangney, B. (2004): Empirical Evaluation of an Adaptive Multiple Intelligence based Tutoring System. In: Proceedings of the 3'rd International Conference on Adaptive Hypermedia and Adaptive Web-based systems (AH'2004). Eindhoven, Netherlands p308-311

[10] Lazaer, D. (1999): Eight Ways of Teaching: The Artistry of Teaching with Multiple Intelligences, SkyLight.

[11] Papanilolaou, K. A., Grigoriadou, M., Kornilakis, H., Magoula, G. D. (2003): Personalising the interaction in a Web-based educational hypermedia system: the case of INSPIRE. User-Modeling and User-Adapted Interaction 13 (3) p213-267

[12] Papankilolaou, K. A & Grigoriadou, M. (2004): Accommodating learning style characteristics in Adaptive Educational Hypermedia Systems. Proceedings of the Workshop "Individual Differences in Adaptive Hypermedia" at the 3'rd International Conference on Adaptive Hypermedia and Adaptive Web-based systems (AH'2004). Eindhoven, Netherlands.

[13] Prensky, M. (2001): *Digital game-based learning*. New York: McGraw-Hill

[14] Rasmussen, K. L. (1998): Hypermedia and learning styles: Can performance be influenced? Journal of Multimedia and Hypermedia, 7(4).

[15] Reigeluth, C.M. (1996): A new paradigm of ISD ? Educational Technology, 36(3)

[16] Riding, R. & Rayner. S, (1997): Cognitive Styles and learning strategies. David Fulton Publishers.

[17] Shearer. C. B. (1996): The MIDAS handbook of multiple intelligences in the classroom. Columbus. Ohio: Greyden Press.

[18] Stern, M & Woolf. B. (2000): Adaptive Content in an Online lecture system. In: Proceedings of the First Adaptive Hypermedia Conference, AH2000

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Pedagogical Agents as Learning Companions: Building Social Relations with Learners

Yanghee Kim
Department of Instructional Technology, Utah State University,
2830 Old Main Hill, Logan, UT 84322, USA

Abstract. This study examined the potential of pedagogical agents as learning companions (PALs) to build social relations with learners and, consequently, to motivate learning. The study investigated the impact of PAL affect (positive vs. negative vs. neutral), PAL gender (male vs. female), and learner gender (male vs. female) on learners' social judgments, motivation, and learning in a controlled experiment. Participants were 142 college students in a computer-literacy course. Overall, the results indicated the interaction effects of PAL affect, PAL gender, and learner gender on learners' social judgments (p < .001). PAL affect impacted learners' social judgments (p < .001) and motivation (p < .05). PAL gender influenced motivation (p < .01) and recall of learning (p < .05). Learner gender influenced recall of learning (p < .01). The implications of the findings are discussed.

Introduction

Educational theorists and researchers often emphasize the importance of the social context of cognition and its applications to learning and instruction. Learning is a highly social activity. Social interaction among participants in learning is seen as the primary source of intellectual development [1]. This emphasis on social cognition seems to demand reframing the conventional use of educational technology and suggests a new metaphor: computers as pedagogical agents.

"Pedagogical agent" refers in general to life-like autonomous characters. In this study, its anthropomorphic nature is emphasized, the purpose being to render personae to computers. Being human-like, a pedagogical agent might build social relations with learners. In particular, pedagogical agents as learning companions (PALs) simulate peer interaction and are designed to take advantage of the cognitive and affective gains of human peer-mediated learning.

PALs should be considered believable realistic virtual peers for building social relations with learners [2]. At the center of believability is PALs' ability to demonstrate affect [3]. Affect, an integral part of social cognition, allows us to successfully function in daily social and intellectual life [4]. Our feelings may signal our judgements and our daily interaction with others. Thus, the affective capability of PALs might facilitate social interaction with learners.

Furthermore, emotion research has indicated the close association of affect and cognition. Affect and cognition are integrally linked to impact on information processing and retrieval [5]. The affective state of a person influences processing style [6]. That is, positive emotions stimulate heuristic, creative, and top-down processing of information, whereas negative emotions stimulate detail-oriented, systematic, and bottom-up processing of information. Also, gender difference manifested in academic interest and cognitive styles becomes more salient in such affective experiences as emotional expression, empathic accuracy, and emotional behavior [7].

This paper addresses several questions: Will the gender/affect interaction in real life be applied consistently to human/computer interaction? In particular, will the gender and affect of a PAL influence a learner's affective and cognitive characteristics as in traditional classrooms? Also, will the impact of a PAL's gender and affect varies depending on a learner's gender? Research has shown human/computer interaction to be consistent with human-to-human interaction [8]. Individuals' emotional experiences are attributed to immediate contexts [9], and so it is highly possible that a PAL's affective states might be transferred to a learner and may influence their information processing, motivation to work with the PALs, and social judgments about the PAL. In this regard, very few studies have been done. Thus, the purpose of the study this paper reports on was to examine the effects of PAL affective expression, PAL gender, and learner gender on learners' social judgements, motivation, and learning.

Method

1. Participants

Participants were 142 undergraduates in a computer-literacy course in a university located in the southeast United States. The participants were novices at the learning task, instructional planning.

2. Materials

2.1. Instructional Module

The instructional module was E-Learn, a web-based environment that introduced instructional planning for e-learning classes. The goal of E-Learn was to introduce basic concepts and proceduresof designing e-learn classes. The module consisted of three phases, Introduction, Goals, and Planning. The students' task in the module was to write their ideas for designing an e-learning class to teach freshmen to be more efficient in time management, depending on the information provided by a PAL. When the participants entered E-Learn, Chris (the PAL) appeared and introduced himself/herself as a peer. As students proceeded, Chris provided context-specific information at each learner's request. All the information provided by the PAL was identical across the experimental conditions. Depending on the conditions, the PALs verbally expressed their affective states. These affective comments were very brief and did not significantly impact total instructional time.

2.2. PAL Design

Male and female PALs, both named Chris, were developed using Poser 5, Mimic Pro 2, and Flash and were integrated into the web-based instructional module. To look peer-like, the PALs were designed to appear approximately twenty years old and wore casual shirts. The PALs' comments were scripted. Given that voice was a significant indicator for social presence [10], voices of male and female college students were recorded. The participants in the study estimated the PALs' age as an average of 20.39 (*SD* = 7.94).

3. Independent Variables

3.1. PAL Affective Expression

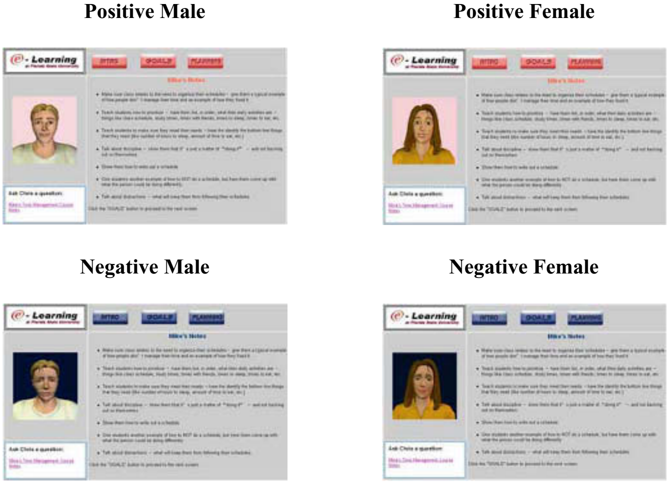
Affective expression was operationalized by verbal and facial expressions, voices, and head movements. Emotion research indicates that people express and perceive emotions mostly through facial expressions, sounds, and body movements, together with verbal manifestations. According to Keltner and Ekman [11], face is the primary source for expressing distinct emotions nonverbally. The distinctive features of individuals' voices also influence how people decipher emotional messages [12]. Body movements too are clearly differentiated according to positive or negative feelings [13]. In addition, Sinclair and colleagues [14] indicate that the color red is interpreted as "upbeat," and fosters *heuristic* processing aligned with positive affect, whereas the color blue is generally interpreted as more depressing and fosters *systematic* processing aligned with negative affect. So the background colors of the module were adjusted to experimental conditions.

The PALs' affective expression had three levels: positive, negative, and neutral. Psychologists typically classify affect as positive if it involves pleasure (e.g., happiness or satisfaction) and as negative if it includes distress (e.g., frustration or anger) [15]. In the positive-affect condition, the PALs had a happy, smiling face and an engaging posture, with eye gaze and with head nodding. The background tone was red. The participants perceived the positive PALs as significantly more "happy looking" than the negative PALs ($p < .001$). In the negative-affect condition, the PALs had a somber and rather frowning face and an aloof posture, with evasive eye contact and less head nodding. The background tone was blue. The participants perceived the negative PALs as significantly more "sad looking" than the positive PALs ($p < .001$). In the Neutral condition, the PALs did not express affect. The background color had a grey tone. Overall, the adjustment of the emotion parameters in the voice/affect editing tool, Mimic Pro 2, operationalized the degree of positive, negative, and neutral expressions of the PALs.

3.2. PAL Gender

Either a male or female version of Chris was included depending on the experimental conditions. The two PALs were identical in all aspects (e.g., comments and emotional expressions), differing only by image and voice. Figure 1 illustrates the PALs with differing affect and gender.

Figure 1. PALs



Positive Male

Positive Female

Negative Male

Negative Female

3.3. Learners' Gender

Learners' gender was a within-group factor. Approximately 40% of the participants were males and 60% females.

4. Dependent Variables

4.1. Social Judgments

Social judgments refered to learners' judgments about the attributes of PALs as their learning partners [16]. Learners' social judgments were measured by a questionnaire consisting of three sub-measures: facilitating (4 items), engaging (4 items), and intelligent (3 items). Items were scaled from 1 (*Strongly disagree*) to 5 (*Strongly agree*). Item reliability in each category was evaluated as coefficient α = .91, .81, and .84 respectively.

4.2. Motivation

Learner motivation was measured by interest. Getzels [17] defines interest as a "disposition organized through experience which impels an individual to seek out particular objects, activities, understandings, skills, or goals for attention or acquisition." Learner interest in the study refered to learners' disposition toward working with the PAL and toward the task. Anderson and Bourke [18] suggest that the range of interest be best expressed on the scale of "interested-disinterested". Learner interest was measured by a questionnaire consisting of three sub-measures: interest in the task (3 items), interest in the PAL (2 items), and desire to work with the PAL (3 items). Items were scaled from 1 (*Strongly disagree*) to 5 (*Strongly agree*). Item reliability in each category was evaluated as coefficient α = .87, .89, and .91 respectively.

4.3. Learning

The author wished to examine the learners' engagement in the interaction with the PAL and speculated that if learners were more engaged, they would recall more of the ideas presented by the PAL. Recall of information and application of the information were regarded as distinct cognitive functions. Thus, learning was measured by the two subcategories of recall and application. In the recall question, students were asked to write all the ideas conveyed by the PALs about designing an e-learning class. The number of legitimate ideas in the students' answers was counted and coded by two instructional designers according to a process suggested by Mayer and Gallini [19]. Inter-rater reliability was evaluated with Cohen's Kappa = .94. In the application question, the participants were asked to write a brief e-learning plan according a given scenario. Students' instructional plans were evaluated by two instructional designers given a scoring rubric scaled 1 (Very poor) through 5 (Excellent). The scoring rubric – which has been used multiple times by Pedagogical Agent Learning Systems Research Laboratory at Florida State University [20] - focused on how specific their plans were in terms of the topic and instructional strategies. Inter-rater reliability was evaluated as Cohen's Kappa = .97.

5. Procedures

The experiment was conducted during a regular session of a computer-literacy course. Participants were randomly assigned to one of the six conditions by PAL affect and gender. The researcher administered the experiment with assistance from the course instructors. The participants first logged on to the web-based E-Learn module by entering demographic information, then performed the task and answered posttest questions. The participants were given as much time as they needed to finish the entire process (approximately 40 minutes, with individual variations).

6. Design and Analysis

The study used a 3 × 2 × 2 factorial design. The variables included PAL affective expression (positive vs. negative vs. neutral), PAL gender (male vs. female), and learner gender (Male vs. Female). For data analyses, three MANOVA's for social judgments, motivation, and learning were first conducted to control for the inflation of family-wise error rates, expected by including multiple dependent measures. Given statistical significance from the overall protected testing, three-way ANOVA's were further conducted for each sub-measure. The significance level for all the analyses was set at $\alpha <$ .05.

Results

1. Social judgments

The overall MANOVA conducted as protected testing indicated an 3-way interaction effect of PAL emotion, PAL gender, and learner gender: Wilks' Lambda = .876, F (6, 240) = 2.97, $p < .001$, partial $\eta^2 = .07$. The MANOVA also indicated the main effect for PAL affective expression: Wilks' Lambda = .76, F (6, 240) = 6.03, $p < .001$, partial $\eta^2 = .13$. To identify the contribution of sub-measures to the overall significance, univariate analyses were further conducted.

For the interaction effect, the univariate results indicated interaction effects on all three sub-measures of facilitating ($p < .01$), engaging ($p < .01$) and intelligent ($p < .05$). When the PALs expressed positive affect, both male and female students rated the male PAL as more facilitating to their learning, more engaging, and more intelligent. However, when the PALs expressed negative affect, male students rated the female PAL as as more facilitating, engaging, and intelligent; whereas female students rated the male PAL as more facilitating, engaging, and intelligent. When the PALs did not express affect (neutral condition), those differences were minimal.

For PAL affective expression, the univariate results revealed significant main effects on "engaging" (F [2, 122] = 12.74, $p < .001$) and on "intelligent" (F [2, 122] = 12.74, $p < .001$). Students who worked with the positive PAL rated the PAL as significantly more engaging and intelligent than students with the negative PAL. Also, students who worked with the neutral PAL rated the PAL as significantly more engaging and intelligent than students with the negative PAL.

2. Motivation

The overall MANOVA revealed the significant main effect for PAL affect (Wilks' Lambda = .87, F [6, 250] = 3.03, $p < .01$, partial $\eta^2 = .07$) and the significant main effect for PAL gender (Wilks' Lambda = .92, F [3, 125] = 3.79, $p < .05$, partial $\eta^2 = .08$). For PAL affect, the univariate results indicated the significant main effect on learners' desire to work with the PAL: F (2, 127) = 4.03, $p < .05$. Students who worked with the positive and neutral PALs desired to keep working with the PALs significantly more than did students who worked with the negative PAL. For PAL gender, the univariate results revealed the main effects on both interest in the PAL (F [1, 127] = 10.04, $p < .01$) and desire to work with the PAL (F [1, 127] = 9.22, $p < .01$). Students of both genders who worked with the male PAL showed significantly higher interest in and desire to work with the PAL than did those who worked with the female PAL.

3. Learning

Learning was measured by two open-ended questions asking recall and application of information. The overall MANOVA revealed the significant main effect for PAL gender (Wilks' Lambda = .83, *F* [2, 59] = 5.99, $p < .01$, partial $\eta^2 = .17$) and the significant main effect for student gender (Wilks' Lambda = .89, *F* [2, 59] = 3.78, $p < .05$, partial $\eta^2 = .11$). For PAL gender, the univariate results indicated a significant main effect on recall: *F* (1, 60) = 6.14, $p < .05$. Students of both genders who worked with the male PAL achieved significantly higher recall scores than did those who worked with the female PAL. For student gender, the univariate results revealed the main effects on recall: *F* (1, 60) = 7.36, $p < .01$. Female students achieved significantly higher recall scores than did male students. Regarding application, there was no significant difference across the groups.

Discussion

The study examined the potential of PAL to build social relations with learners by implementing PAL affect and gender. To do so, the impact of PAL affect, PAL gender, and learner gender was investigated in terms of learners' social judgments, interest, and learning. Overall, the study revealed the interaction effects of PAL affect, PAL gender, and learner gender on learners' social judgments, to reflect human-to-human relations. PAL affect and gender influenced learner interest in working with PALs. The gender of PAL and learner had influence on recall of learning.

The study was grounded in human emotion research revealing the close interaction between gender and emotion in human relationship. Similarly, the results revealed that affect and gender were significant indicators for learners' social judgments in the PAL-based environment. Also, the PAL's positive affect had an positive impact on learners' social judgments and motivation. Specifically, students who worked with the PAL that expressed positive affect rated the PAL as significantly more engaging and more intelligent and more desirable to work with than did students who worked with the negative PAL. These results were consistent with classroom research indicating that students in classrooms placed value on having teachers who showed positive affect [21] and that teachers' expressions of negative emotions were less favorable and associated with learners' negative affect [22].

Regarding PAL gender, students who worked with the male PAL showed higher interest in and desire to work with the PAL. This positive motivation might lead them to engage in and recall the PAL's comments more than those who worked with the female PAL. This superior impact of the male PAL to the female counterpart is analogous to the previous study indicating learners' high motivation toward and favorable perceptions of male pedagogical agents over female agents [23]. This tendency implies that stereotypic expectations of males and females in human relationships [24] might be infused to PAL/learner relationships. In the future, it will be worthwhile to examine ways to reduce stereotypic bias associated with gender by manipulating PAL gender along with other characteristics of learners and PALs in various learning contexts .

Regarding learner gender, female students showed higher recall scores than did male students, perhaps because the female students tended to show positive attitudes toward the PALs in general, indicated by their higher ratings on most of the items. This trend was also observed in previous studies [23, 25]. This positive attitudes of female students might lead them to engage more fully in the task and, consequently, acquire and recall more information.

In the current study, however, there were some limitations. First, learners' social judgments were not differentiated across the PAL who expressed positive affect and the PAL who did not express affect. Perhaps because the individual PALs' emotional expressions did not vary--all happy, all sad, or all neutral--some students might not have been aware of PAL affect while working in the instructional module unless the affect was clearly negative. This speculation sounds persuasive, since the awareness of feelings mediates the effect of feelings on social judgments [16]. Second, the study was done by one-time implementation. Building social relations with learners may require sustained interactions in a longer term. Also, the study was focused on an "outer" quality of the PALs and may serve as a preliminary step for the investigation of PALs performing intelligently. Future research might overcome the limitations of the current study.

References

[1] V. John-Steiner and H. Mahn, "Sociocultural contexts for teaching and learning," in *Handbook of psychology: Educational psychology*, vol. 7, A. Reynolds, M. William, and G. E. Miller, Eds. New York: John Wiley and Sons, 2003, pp. 125-151.

[2] K. Dautenhahn, A. H. Bond, L. Canamero, and B. Edmonds, "Socially intelligent agents: Creating relationships with computers and robots." Norwell, MA: Kluwer Academic Publishers, 2002.

[3] J. Bates, "The nature of characters in interactive worlds and the Oz project," School of Computer Science, Carnegie Mellon University, Pittsburgh, PA CMU-CS-92-200, 1992.

[4] R. Adolphs and A. R. Damasio, "The interaction of affect and cognition: A neurobiological perspective," in *Feeling and Thinking: The Role of Affect in Social Cogniton*, J. P. Forgas, Ed.: Cambridge University Press, 2000.

[5] G. H. Bower and J. P. Forgas, "Mood and social memory," in *Handbook of Affect and Social Cognition*, J. P. Forgas, Ed. Mahwah, NJ: Lawrence Erlbaum Associates, Inc., 2001.

[6] N. Schwarz, "Situated cognition and the wisdom in feelings," in *The wisdom in feelings*, L. F. Barrett and P. Salovey, Eds. New York: The Guilford Press, 2002, pp. 145-166.

[7] L. Brody, *Gender, emotion, and the family*. Messachusetts: Harvard University Press, 1999.

[8] B. Reeves and C. Nass, *The Media Equation: How people treat computers, television, and new media like real people and places*. Cambridge: Cambridge University Press, 1996.

[9] C. Saarni, "Emotion communication and relationship context," *International Journal of Behavioral Development*, vol. 25, pp. 354-356, 2001.

[10] R. E. Mayer, K. Sobko, and P. Mautone, "Social cues in multimedia learning: role of speaker's voice," *Journal of Educational Psychology*, vol. 95, pp. 419-425, 2003.

[11] D. Keltner and P. Ekman, "Facial expression of emotion," in *Handbook of Emotions*, M. Lewis and J. M. Haviland-Jones, Eds. New York: The Guilford Press, 2000, pp. 236-249.

[12] J. Bachorowski and M. J. Owren, "Vocal acoustics in emotional intelligence," in *The wisdom in feelings*, L. F. Barrett and P. Salovey, Eds. New York: The Guilford Press, 2002, pp. 11-36.

[13] M. Chen and J. A. Bargh, "Consequences of automatic evaluations: Immediate behavioral predispositions to approach or avoid the stimulus," *Personality and Social Psychology Bulletin*, vol. 25, pp. 215-224, 1999.

[14] R. C. Sinclair, A. S. Soldat, and M. M. Mark, "Affective cues and processing strategy: Color coded forms influence performance," *Teaching of Psychology*, vol. 25, pp. 130-132, 1998.

[15] V. Ottati, N. Terkildsen, and C. Hubbard, "Happy faces elicit heuristic processing in a televised impression formation task: A cognitive tuning account," *Personality and Social Psychology Bulletin*, vol. 23, pp. 1144-1156, 1997.

[16] G. L. Clore, "Affective influences on social information processing," in *Handbook of Affect and Social Cognition*, J. P. Forgas, Ed. Mahwah, NJ: Lawrence Erlbaum Associates, Inc., 2001.

[17] J. W. Getzels, "The problem of interests: A reconsideration," *Supplementary Education Monographs*, vol. 66, pp. 97-106, 1966.

[18] L. W. Anderson and S. F. Bourke, "Assessing affective characteristics in the schools," 2nd ed. Mahwah, NJ: Lawrence Erlbaum Associates, 2000.

[19] R. E. Mayer and J. K. Gallini, "When is an illustration worth ten thousand words?" *Journal of Educational Psychology*, vol. 82, pp. 715-726., 1990.

[20] Y. Kim and A. L. Baylor, "Pedagogical agents as learning companions: The role of competency and type of interaction," *Educational Technology Research & Development*, in press.

[21] C. A. Wong and S. M. Dornbusch, "Adolescent engagement in school and problem behaviors: The role of perceived teacher caring," presented at Annual Meeting of the American Educational Research Association, New Orleans, LA, 2000.

[22] R. Lewis, "Classroom discipline and student responsibility: The students' view," *Teacher Education*, vol. 17, pp. 307-319, 2001.

[23] A. L. Baylor and Y. Kim, "Pedagogical agent design: The impact of agent realism, gender, ethnicity, and instructional role," presented at Intelligent Tutoring Systems, Maceió, Alagoas, Brazil, 2004.

[24] L. L. Carli, "Gender and social influence," *Journal of Social Issues*, vol. 57, pp. 725-741, 2001.

[25] Y. Kim, "Learners' expectations on the desirable characteristics of learning companions," presented at the Annual Conference of American Educational Research Association, San Diego, CA, 2004.

Acknowledgements

This work was sponsored by National Science Foundation Grant # IIS-0218692 and the Pedagogical Agent Learning Systems lab at Florida State University. The author thanks Dr. Amy L. Baylor for the support.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

The Evaluation of an Intelligent Teacher Advisor for Web Distance Environments

Essam KOSBA, Vania DIMITROVA, Roger BOYLE
School of Computing, University of Leeds, UK
E-mail: {essamk, roger, vania}@comp.leeds.ac.uk

Abstract. There is a lack of automatic features to help instructors to effectively manage distance courses delivered with Web Course Management Systems (WCMS). We have developed a system named TADV (Teacher ADVisor) that uses student tracking data to build fuzzy student, group, and class models to generate advice that highlights important situations to instructors and recommends feedback to be sent to students. This paper presents an evaluative study which shows that TADV provides practical and effective advice valued by both teachers and students. The instructors became more knowledgeable about what was happening in the distance class and were able to send regular feedback to the students. There was improvement in the students' overall satisfaction and their link with instructors.

1. Introduction

WCMS are widely used to deploy distance courses. The instructors, who play a central role in managing such courses, need to have a good understanding of what the students' needs and problems are. Recently, intelligent techniques have been used to enhance WCMS [1] but, in line with most AIED systems (e.g. [2]), the effort is focused mainly on providing adaptive help to students. There is a lack of automatic features to guide instructors by pointing at important situations and highlighting possible problems. Such features may help instructors overcome distance learning problems like student isolation and disorientation, and reduce the workload and communication overhead needed for managing distance classes effectively [3] and [4]. Our research focuses on providing appropriate advice to help facilitators manage courses delivered via WCMS effectively. Similarly to [5] and [6], we adopt AI methods to assist teachers in learning environments. We have developed a Teacher ADVisor (TADV) framework [7] which uses student tracking data to build fuzzy student, group and class models [8], based on which advice is generated and provided to facilitators [9]. A TADV prototype was developed to extend a conventional WCMS [7].

This paper presents an empirical study to evaluate TADV in realistic settings. We will estimate the strengths and weaknesses of the approach to facilitate the development of TADV in similar advisory systems. To the best of our knowledge, there are no evaluative studies geared towards measuring the benefits of advising instructors in distance education. Comprehensive empirical evaluations of adaptive systems are hard to find due to short development cycle and difficulties to measure the outcomes [10]. Evaluation in distance learning settings is even more difficult for the absence of standards, high costs, and scarcity of expertise, among others. Based on existing evaluative studies in distance learning [11] and in AIED [12], we combined quantitative and qualitative data in a control-group study to examine the suitability of advice and the benefits for both instructors and students. Next, the paper will briefly introduce the TADV (#2) and outline the evaluative study (#3). The following sections, will discuss the results from the study focusing on the suitability of advice (#4), benefits for facilitators (#5), and benefits for students (#6). In the conclusions, we will point at our plans for future work.

2. The TADV System

Figure 1 shows the TADV architecture. **PART-I** represents the conventional procedure performed to build and use a WCMS course. *Domain Knowledge Base* contains course material usually prepared by the teacher. The tracking data where WCMS records the students' interactions in the course are recorded in *Student Data Base*.

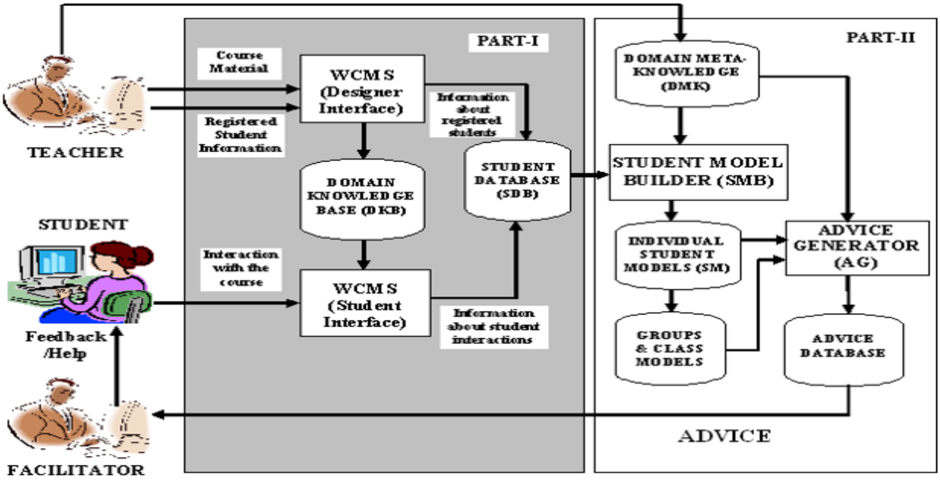

PART-I
PART-II
TEACHER
Course Material
Registered Student Information
WCMS (Designer Interface)
Information about registered students
DOMAIN KNOWLEDGE BASE (DKB)
STUDENT DATABASE (SDB)
STUDENT
Interaction with the course
WCMS (Student Interface)
Information about student interactions
Feedback /Help
FACILITATOR
DOMAIN META-KNOWLEDGE (DMK)
STUDENT MODEL BUILDER (SMB)
INDIVIDUAL STUDENT MODELS (SM)
ADVICE GENERATOR (AG)
GROUPS & CLASS MODELS
ADVICE DATABASE
ADVICE

Figure 1. The TADV Architecture. PART II represents the TADV extension to conventional WCMS.

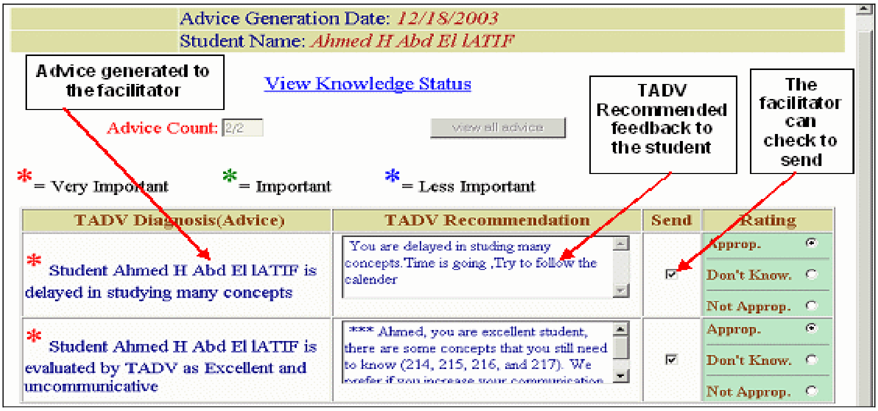
PART-II represents the TADV architecture. The *Domain Meta-Knowledge* (DMK) possesses the information that describes the course material. It contains metadata about the course, course calendar, lessons, concepts and how they are related, and content material. TADV follows the IEEE LOM metadata standards [13] together with some additional attributes required for the adopted fuzzy student modeling approach [8].

TADV includes three levels of student modeling: individual *Student Models* (SMs), *Group Models* (GMs), and *Class Models* (CMs). SM contain: Student Profile, Student Behavior Model (keeps student's learning sessions and interactions and detailed information of his/her activities), Student Preferences (e.g. student's preferred types of learning objects), and Student Knowledge Model (student's level of understanding of course concepts). The main source for modeling students is the tracking data generated by WCMS. An overlay approach is used to represent knowledge status in SM, GM and CM. In SM, each concept is associated with a measurement of the student's knowledge status in relation to that concept. Similarly, GM and CM overlay the domain concepts with an aggregate measurement of the knowledge status of all students in the group or class ([7] gives detail).

The *Advice Generator* (AG) uses a set of predefined conditions to specify advising situations that are associated with advice templates. Each situation is defined by: Stimulating Evidence (that triggers the situation); Investigated Reason (that provides evidence and is based on information from the SM, GM, and CM); Teacher advice template (used to compile advice to the facilitator); Recommended feedback template (used to generate suggestions of what the teacher can send to the student, group, or class). When the AG recognizes a situation, the corresponding templates are activated to generate advice to the teacher and, when available, a recommendation of what can be sent to the student.

TADV follows an advice taxonomy that includes advice concerning individual student performance (Type-1), group performance (Type-2), and class performance (Type-3). The TADV advice generation mechanism is presented in detail in [9].

TADV was integrated in the Centra WCMS. The TADV extension followed the architecture in Figure 1 and was implemented on Microsoft SQL Server 2000 and Active Server Pages (ASP) technology with ODBC (Open Data Base Connectivity) drivers. Visual Basic and Java scripts are used as development languages. Figure 2 shows part of the facilitator's interface, while Figure 3 shows feedback to students from the evaluative study.



Advice Generation Date: 12/18/2003

Student Name: Ahmed H Abd El lATIF

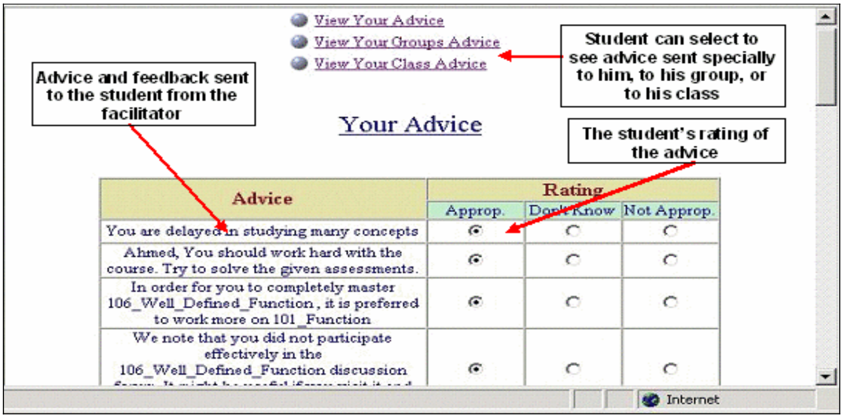
Advice generated to the facilitator

TADV Recommended feedback to the student

The facilitator can check to send

TADV Diagnosis(Advice)	TADV Recommendation	Send	Rating	
* Student Ahmed H Abd El lATIF is delayed in studying many concepts	You are delayed in studing many concepts.Time is going ,Try to follow the calender	☑	Approp.	●
			Don't Know.	○
			Not Approp.	○
* Student Ahmed H Abd El lATIF is evaluated by TADV as Excellent and uncommunicative	*** Ahmed, you are excellent student, there are some concepts that you still need to know (214, 215, 216, and 217). We prefer if you increase your communication	☑	Approp.	●
			Don't Know.	○
			Not Approp.	○

Figure 2. A screen used to display advice to the facilitator along with recommended feedback that can be sent to the student. The facilitator can modify the recommended advice before sending it and can choose either to send or suppress it. The rating section is for evaluation purposes.



Advice and feedback sent to the student from the facilitator

Student can select to see advice sent specially to him, to his group, or to his class

The student's rating of the advice

Advice	Rating		
	Approp.	Don't Know	Not Approp.
You are delayed in studying many concepts	●	○	○
Ahmed, You should work hard with the course. Try to solve the given assessments.	●	○	○
In order for you to completely master 106_Well_Defined_Function , it is preferred to work more on 101_Function	●	○	○
We note that you did not participate effectively in the 106_Well_Defined_Function discussion forum. It might be useful if you visit it and	●	○	○

Figure 3. A screen displays advice to a student, i.e. what the teacher has sent to this student. The rating section is for evaluation purposes.

3. The Evaluative Study

The TADV evaluation aimed at verifying the usability and functionality of the system and examining the benefits of the approach for facilitators and students. It comprised a formative and a summative phase [14]. The formative evaluation focused on the system performance and included several students and teachers whose comments and suggestions were used to improve the system for the summative phase presented here.

The summative evaluation examined the benefits of the approach by integrating TADV within a distance learning environment in a Discrete Mathematics course at the Arab

Academy for Science and Technology (AAST), Alexandria, Egypt. Three facilitators and 30 students took part in the study. Due to limitations imposed by the university administration, TADV was used three weeks and only for two topics of the course (Functions and Relations) the other topics were taught in traditional face-to-face lectures. The students were divided into two groups: *control group* (*Class-1*) where the students worked with TADV via distance, the system built models for them but the advice generation was suppressed, consequently, the facilitators were not advised (i.e. students in this group experienced traditional use of WCMS and got feedback from facilitators through discussion forums and e-mail); *experimental group* (*Class-2*) where the students worked with TADV via distance, system built models for them, generated advice to the facilitators (same facilitators of the control group) who then sent feedback to the students. The group allocation ensured equal distribution of student knowledge, academic background, gender, and nationality.

Examples of advice generated during the study are presented in Table 1 to illustrate the information the facilitators were given about cognitive, behavioral, and social aspects of the students, and how advice helped the facilitators to compose feedback to the students. During the study, extensive data was collected, including log files, pre and post test, teacher interviews and observations, and student questionnaire. The results are shown next.

Table 1. Sample of the advice generated during the experimental study. "***" means that the recommended feedback is composed by the facilitator based on the information provided by TADV.

Advice to the facilitator	Recommended feedback to the student	Explanation and Results
Student Ahmed Othman is delayed in studying many concepts.	You are delayed in studying many concepts. Time flies. Try to follow course calendar.	TADV found that the student is delayed in studying several concepts and sent this information to the facilitator. He sent feedback to the student who was encouraged to follow the course calendar.
Student Ahmed Abdel Latif is evaluated by TADV as Excellent and uncommunicative.	*** Well done Ahmed, try to help your peers.	TADV found that the student was excellent but he was uncommunicative. The facilitator tried to motivate the student to become communicative.
Student Mostafa El Shami is evaluated by TADV as Weak and uncommunicative.	*** You should work hard with the course. Try to solve the given assessments. You should also communicate with your peers through the discussion forums.	TADV found that the student was weak and uncommunicative. The facilitator was advised to motivate the student. He composed and sent the shown feedback.
Student Mostafa El Shami should be advised to study Identity.	In order for you to master Composition and Identity, it is highly recommended to study Identity first.	TADV found that the student was struggling with the concept Composition and Identity because this concept was strongly related to Identity which was still unlearned by the student. The facilitator realized that the student was struggling with both concepts and sent the feedback to the student.
TADV can not evaluate Group1 because most of its members have not started the course yet.	*** For the group members who did not start the course, time is going, please start the course as soon as possible.	TADV found that most of the Group1 members did not start the course. The facilitator became knowledgeable about the problem and composed the shown feedback to the group members.
Group2 is evaluated by TADV as Weak and uncommunicative group.	*** All members of Group2 should work more effectively with the course. Try to solve the assessments and communicate with your peers in the group through the discussion forums and e-mail.	TADV found that Group2 was weak and uncommunicative. The facilitator became more knowledgeable about this group and composed the feedback to motivate the group members.
Shady Nossier, Ahmed Abd El Latif are the most excellent students relative to the whole class, while Amr Ismail, Abd Elrahman Gabr, and Mohamed Abdel Aziz are the weakest students.	*** There are many students who did not start working with the course. Please, those students should start the course as soon as possible. Students who face problems can communicate with Shady Nossier and Ahmed Abdel Latif; they are excellent.	TADV informed the facilitator about the most excellent and weakest students in the class. The facilitator read all advice generated about the class not just the shown one. He then composed the shown feedback and sent to everybody to motivate them to actively work on the course. He encouraged the students to contact the excellent ones.

4. Suitability of Advice Types

Examining the suitability of advice was needed to validate the whole framework. Suitability of advice was measured by considering what the facilitators thought about the advice features, how they evaluated the generated advice, what advice they sent to their students and how the students evaluated the feedback they received. Table 2 shows the number of advice pieces generated to facilitators and sent to students and the results of advice rating with respect to each advice type. The ratings, as well as some questions from the teacher interviews and the student questionnaire, were used to analyze the suitability of advice. Due to space limit, we will only summarize the analysis of the findings (a comprehensive description is given in [7]):

Table 2. Number of advice pieces generated to facilitators and sent to students and advice rating by facilitators and students according to advice type. (A: Appropriate, D: Do not know, N: Not Appropriate)

Advice Type	No. of Advice	Facilitator Rating			Sent Advice	Student Rating		
		A	D	N		A	D	N
1-1 (student's knowledge status	348	188	156	4	189	57	35	6
1-2 (Student's Delays)	52	52	0	0	50	40	0	0
1-3 (Weak students)	47	47	0	0	45	33	0	0
1-4 (Excellent Students)	6	6	0	0	5	3	1	0
1-5 (Student not started the course)	82	82	0	0	77	6	0	0
Total Type-1 (Related to individual students)	**535**	**375**	**156**	**4**	**366**	**139**	**36**	**6**
2-1 (Group knowledge status)	144	32	95	0	16	21	9	3
2-2 (Weak group)	3	3	0	0	3	7	0	0
2-3 (Excellent group)	0	0	0	0	0	0	0	0
2-4 (Group members not started the course)	11	11	0	0	11	3	3	1
Total Type-2 (Related to groups of students)	**158**	**46**	**95**	**0**	**30**	**31**	**12**	**4**
3-1 (Class knowledge status)	104	71	33	0	1	5	5	0
3-2 (Excellent/weak students relative to class)	7	6	1	0	2	14	7	0
3-3 (Communicative students relative to class)	7	6	1	0	0	0	0	0
3-4 (Active and Inactive students)	7	6	0	1	0	0	0	0
3-5 (Class members did not start the course)	5	5	0	0	5	11	10	2
Total Type-3 (Related to the whole class)	**130**	**94**	**35**	**1**	**8**	**30**	**22**	**2**
Total ALL	**823**	**515**	**286**	**5**	**404**	**200**	**70**	**12**

- Facilitators were satisfied with the advice generated by TADV regarding advice types, contents, and the situations addressed. The facilitators appreciated the generated advice and agreed that it was needed and useful.
- Students found that advice was suitable and guided. They regarded as most helpful the feedback that pointed out the delayed or struggling students. Some students asked for advice to be generated on a daily basis and others suggested the advice to be in Arabic.
- Type1-2 (student delays), Type1-5 (student did not start the course), Type2-4 (most group members did not start the course), and Type3-5 (most class members did not start the course) was regarded as appropriate and helpful by both teachers and students. This shows the importance of advice related to students' behavior with the course.
- The appropriateness of Type1-3 (Weak student), Type1-4 (Excellent student), Type2-2 (Weak group), and Type3-2 (Excellent and Weak students relative to the class) show the importance of the automatic student evaluation mechanisms for the facilitators.
- The study showed the appropriateness and the importance of the advice types related to students' knowledge status [Type1-1 (student knowledge status), Type2-1 (group knowledge status), and Type3-1 (class knowledge status)]. However, for these types of advice the facilitators stressed the issue of reducing the pieces of advice in some situations (e.g. when a student was struggling with many concepts, the corresponding number of advice pieces were generated, while the teachers preferred one piece of advice to highlight that this student was struggling with the course concepts). This shows the need to add some advice filtration and aggregation mechanisms.

5. Benefits for Facilitators

TADV is directed towards helping facilitators to appropriately manage their distance classes through providing them with important information about the behavior of their distant students. The facilitators' feedback was considered as crucial part in evaluation. It was gathered during advice generation sessions and via a group interview at the end.

The facilitators thought that although the study time was limited, they felt that by using TADV as a framework for Web-based learning it was possible to achieve similar learning gains to what would have been achieved in a face-to-face learning environment. However, the facilitators pointed out that the learning gains could not be attributed solely to the interaction with TADV because some students did not use the available learning objects and others used TADV just to solve the available assessment quizzes. This is valid for all online distance education environments in which students can freely study on their own using the online material, printed material, textbooks, or any supplementary materials they find. It is difficult to isolate the effect of the learning environment. The facilitators regarded as very positive the fact that using TADV they became aware of the cognitive status and behavior of their distant students, as one of them commented:

> "*Class-2 seems clear to me - I can easily know who is delayed, who did not start the course, who is good and who is weak. I can also know what concepts students are struggling with.*"

Through the generated advice facilitators became aware of the following issues:

- Problems with individual students, groups, and whole class, e.g. what concepts students were struggling with.
- Students' behavior – who followed the course calendar, who was delayed, who was starting study just before the course ends, and who did not start the course.
- Students' knowledge status as judged by the system – how the students were progressing with the course material and what their communication status was.

It was difficult to compare the teachers' communication overload that resulted from both classes because the number of exchanged e-mails was very limited (the facilitator of Class-1 received 6 e-mails, while the facilitator of Class-2 received only 2). The limited number of e-mails can be explained by the short experimental time and by the students' unfamiliarity with using e-mails to make contact with their teachers. This points at some cultural differences that may have to be considered in analyzing student behavior which was not considered in TADV and would require further studies.

It was also important to analyze the time the facilitators spent during the advising sessions in order to examine whether any additional overload was added to the facilitators, or not. Table 3 shows, for each of the seven sessions, the amount of advice, and the session time (the time facilitators spent to read, compose, and send feedback). The average time (53 minutes) per session included the time spent on reading advice, editing/composing feedback, rating each advice for appropriateness, and discussing aspects asked by the experimenter.

Table 3. Times spent in the advising sessions.

Advising session	No. of advice pieces	Advising session time (Minutes)
1	41	55
2	29	40
3	33	35
4	45	50
5	55	45
6	173	65
7	482	85

This demonstrates that advising sessions did not consume much of the facilitators' time, especially if compared to the online chatting sessions, which require much longer time to handle and are difficult to arrange, especially when students are from areas with different time zones. Furthermore, the time the facilitators would have had to spend in order to gain understanding of their students – which they achieved with TADV – by using only the monitoring features provided by WCMS would have been much longer.

6. Benefits for Students

We have to acknowledge that within the short period of the experimental study, it was not realistic to expect a significant enhancement in the students' learning gains and their affective aspects. Nevertheless, we have been able to collect data that shows some potential benefits for students. Following are the most important outcomes concluded from the analysis of students' questionnaire, and pre-test and post-test scores:

- The percentages of students who thought that working with TADV was worse than face-to-face lecture was 62% in Class-1 against only 29% in Class-2. This might be attributed to the availability of the advice and feedback from the facilitators, which was the only differentiating factor between the two classes. The students felt the connection with the facilitator and appreciated the regularity of the feedback.
- The students were interested to know how they were evaluated by their facilitators. This stresses the students' need to receive feedback and get help from their teachers, which, in turn, shows the importance of providing support to teachers to give appropriate feedback to the students.
- Most students in Class-2 (62%) felt that they were continuously guided by the facilitators. Hence, using TADV led to forming the students' impression that the facilitators supervised them during the distance course. This can be linked to reducing the chance of being isolated and lost in the course.
- The availability of the advice reduced the students' need to contact their teachers.
- The level of student satisfaction with regard to the contact they had with the facilitator was higher in Class-2 (23% in Class-1 compared to 54% in Class-2). The students' satisfaction with the contact they have with their teachers is important for lessening the students' feeling of isolation in distance learning.
- Regarding the students' overall satisfaction, Class-2 responses were more positive than Class-1 responses regarding issues like enjoyment (31% in Class-1 vs. 77% in Class-2), self esteem (38% in Class-1 vs. 71% in Class-2), and recommending the course to other students (42% in Class-1 vs. 71% in Class-2).

All students who participated in the evaluation study completed a pre-test and a post-test. Pre-test scores were used as an indication of the students' learning levels gained from face-to-face teaching prior to the experimental study. Two statistical techniques were used for this analysis – *t-test* and *Effect Size*. The analysis showed the following results:

- For [df (degree of freedom) = 28, t = 2.763, α (probability of error) = 1% i.e. 99% confidence level] there was no significant difference in the pre-test scores of the two classes, as well as between the General Point Average grades. This indicates similarity between the control and experimental group.
- For [df = 28, t = 2.763, α = 1%] there was no significant difference between the post-test scores, i.e. there was no significant effect on post-test scores due to the availability of advice/feedback directed to Class-2 students. This result was expected due to the short time of the experimental study.
- Effect size was applied to the participants in both classes to evaluate whether the students' learning gain differed when using TADV with advising features. There was a small improvement in learning gains for the students of Class-2 (effect size = 0.288). It is important to acknowledge that this small improvement cannot be attributed firmly to the availability of TADV advising features.

6. Conclusion

This research is a step toward increasing the effectiveness of distance education with WCMS platforms through the use of Artificial Intelligent techniques to support teachers. Our research contributes to a recently emerging trend for incorporating intelligent techniques in WCMS. We have demonstrated an approach of using student tracking data to implement features that extend the functionality of traditional WCMS to support teachers. The essence of our approach is the building of student, group, and class models and the use of these models to generate advice to help teachers get a better understanding of their students. The paper briefly presented the TADV system and discussed the results from an evaluative study in realistic settings.

The empirical study has shown that TADV provides practical and effective advice. It allows advice generation and informing of instructors, which, in turn, made it easy to send help and feedback to distance students. The instructors confirmed the appropriateness of the generated advice and appreciated the knowledge they gained about their students. The students appreciated the feedback received from the instructors, which was a result of TADV recommendations. The study showed better overall satisfaction and social aspects for the students who used TADV advising features.

Our immediate future plans include adding algorithms for filtering and aggregation of advice and implementing a TADV component within AASTOLP - a bespoke course management system being built in the AAST. Employing TADV in AASTOLP will give us the chance to use Arabic language in the advising features in addition to English language and to deploy TADV in real, long-term settings involving a significant number of students and faculty. This will enable larger studies to further examine benefits and pitfalls of the TADV approach. In the long run, we consider incorporating open student models to improve the student modeling reliability in TADV. Further studies are needed to consider cultural diversity in student behavior and the corresponding advice generated by TADV.

References

[1] Calvo, R. & Grandbastien, M. (Eds.) (2003). Intelligent Management Systems Workshop. *Supplementary Proceedings of AIED'2003*, Sydney, Australia.

[2] Brusilovsky, P. (1999). Adaptive and Intelligent Technologies for Web-Based Education. In C. Rollinger & C. Peylo (Eds.) *Künstliche Intelligenz*, Special Issue on Intelligent Systems and Tele-teaching, 4, pp. 19-25.

[3] Galusha, J. (1997). Barriers to Learning in Distance Education. *Interpersonal computing and technology: An electronic journal for the 21st century*, 5(3/4): pp. 6-14. http://www.infrastruction.com/barriers.htm.

[4] Rivera, J. & Rice, M. (2002). A comparison of Student Outcomes and Satisfaction between Traditional and Web-based Course Offerings. *Journal of Distance Learning Administration*, 5(3).

[5] Delozanne, E., Grugeon, B., Prévit, D., & Jacoboni, P. (2003). Supporting Teachers When Diagnosing Their Students in Algebra. In É. Delozanne, & K. Stacey (Eds.), Workshop *Advanced Technologies for Mathematics Education*, *Supplementary Proceedings of AIED*, Sydney, IOS Press, Amsterdam, pp. 461-470.

[6] Merceron, A., & Yacef, K. (2003). A Web-Based Tutoring Tool with Mining Facilities to Improve Learning and Teaching. In U. Hoppe, F. Verdejo, & J. Kay (Eds.), *Proceedings of the 11th International Conference on Artificial Intelligence in Education*, Sydney, Australia, IOS Press, pp. 201-208.

[7] Kosba, A. (2004). Generating Computer-Based Advice in Web-Based Distance Education Environments. PhD thesis, School of Computing, University of Leeds, UK, submitted.

[8] Kosba, E., Dimitrova, V., and Boyle, R. (2003). Fuzzy Student Modeling to Advise Teachers in Web-Based Distance Courses. *International Journal of Artificial Intelligence Tools*, Special Issue on AI Techniques in Web-Based Educational Systems, World Scientific Net, 13(2): pp. 279-297.

[9] Kosba, E., Dimitrova, V., & Boyle, R. (2005). Using Student and Group Models to Support Teachers in Web-Based Distance Education. *Proceedings of UM05*, to appear.

[10] Weibelzahl, S., & Weber, G. (2003). Evaluating the Inference Mechanism of Adaptive Learning Systems. In Brusilovsky, P., Corbett, A. & de Rosis, F. (Eds.) *User Modeling: Proceedings of the 9th International Conference*. Lecture Notes in Computer Science; Springer-Verlag, Berlin, pp. 154-168.

[11] Hara, N., & Kling, R. (1999). Students' Frustrations with a Web-Based Distance Education Course. *First Monday – Peer-Reviewed Journal on Internet*, Vol. 4, No. 12. http://firstmonday.org/issues/issue4_12/hara

[12] Ainsworth, S. (2003). Evaluation Methods for Learning Environments. Workshop held in conjunction with the 11th International Conference on Artificial Intelligence in Education AIED2003, Sydney, Australia.

[13] H. Wayne, et al, Draft Standard for Learning Object Metadata (Final Draft Document IEEE 1484.12.1), Copyright © 2002 by the Institute of Electrical and Electronics Engineers, Inc.

[14] Mark, M. & Greer, J. (1993). Evaluation Methodologies for Intelligent Tutoring Systems. *Journal of Artificial Intelligence and Education*, 4(2/3), pp. 129-153.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

A Video Retrieval System for Computer Assisted Language Learning

Chin-Hwa Kuo, Nai-Lung Tsao, and Chen-Fu Chang
Department of Computer Science and Information Engineering
Tamkang University, Taiwan

David Wible
English Department, Tamkang University, Taiwan

Abstract. Video has been presented as an effective medium for computer assisted language learning. However, there exist few efficient accessing and managing tools. In this paper, we propose a retrieval system for a large video database. The system retrieves video clips by searching video subtitle text. For language learning, we have designed a syntax search engine embedded in this system. This search engine uses regular expression as the query language and an index construction algorithm is well-designed for speeding up regular expression matching. To ease the burden of authoring lessons from these materials, we implement an automatic video segmentation algorithm to present complete events or actions as final results. The integration of this system and other tools in our authoring environment is also briefly described.

1. Introduction

Video has been presented as an effective medium for computer assisted language learning [1,2]. This medium provides features which are very beneficial to language learning. To name just a few, first, videos, such as entertaining movies, are attractive to learners. Digital videos provide authentic daily-life conversations by native speakers, which is more realistic than those specify-design teaching materials. Digital video also can support language learning by providing listening comprehension training.

Currently, current ways of accessing digital video are very inefficient in light of the purposes they could serve in language learning. Therefore the results of digital video in learning are limited. Common usage of videos in classroom is as follows. When teachers show movies or other types of digital videos in class, most of them usually request their students to perform some movie-related activities after watching the movies (e.g. theme-based discussion or listening comprehension training). Yet digital videos can be used in many others ways. For example, if a teacher would like his/her students to learn how to use the word "apology," he or she would want to have video clips where this word is being used and integrate these into a learning flow with many examples including the word "apology." To obtain good examples of the correct usage of the word, teachers need to review these digital videos one by one and record those suitable clips. This kind of operations is tedious and time consuming. Apparently, in most cases, we only need a small portion of the whole movie.

Therefore, in this paper, we propose a retrieval system for a large video database to support the above mentioned scenarios. Since the most valuable part of digital video in language learning is the context of the conversation, content-based video retrieval does not need to be implemented in our system. In our approach, the subtitle text of digital videos collected in the system have been extracted sentence by sentence and have been indexed with their occurring time. To match the application end, we can accept two kinds of query inputs, namely, keyword-based query and syntax-based query, for searching these subtitle texts. Syntax-based query is implemented by a regular-expression-based (regex-based) search engine which is embedded in our system. In order to achieve near-real-time response, this search engine is designed with a special *index construction* and a *query processing* mechanism. The search results are sentences which are distributed throughout the video database. At the end of this stage, the search result is not practical because it contains no context. Providing an editing tool to extend the clip seems like a good way but it is time consuming. For this reason, this system provides an option to present the result by a "complete scene", in which clip provided from the digital video presents a complete action or event but sometimes it depends on the filming manner. To prevent automatic scene detection inaccuracies, the editing tool still allows the user to override the automated segmenting result and extend the clip by time, sentence, and scene unit.

Although the regex-based search engine is efficient, it is unreasonable to expect common users of our system, English teachers or learners, to master the use of regular expressions. Consequently, a user-friendly query interface is necessary. We designed a user interface, called regex query generator, for bridging the gap. The regex query generator transfers user input into a regex query. These details are given in Section 2.

Although teachers now have a powerful tool to get what they want to show their students, they still need an editing environment to arrange the flow of their teaching content and the location of these video clips and subtitle text. We built an Application Programming Interface (API) for integrating our system into other authoring environmens. We will show the integration of the result of the designed video retrieval system and the authoring tool of IWiLL [3]. The authoring environment then provides the ability to embedding video clips and subtitle text into teaching materials.

The rest of the paper is organized as follows: in Section 2, we will present an overview of the system architecture. The detailed index construction and query processing approaches of regex-based search engine are proposed and the video segmentation technique which we applied is described in Section 3. The integration of the system and IWiLL authoring tool is shown in Section 4. Section 5 presents the conclusion and future works.

2. System overview

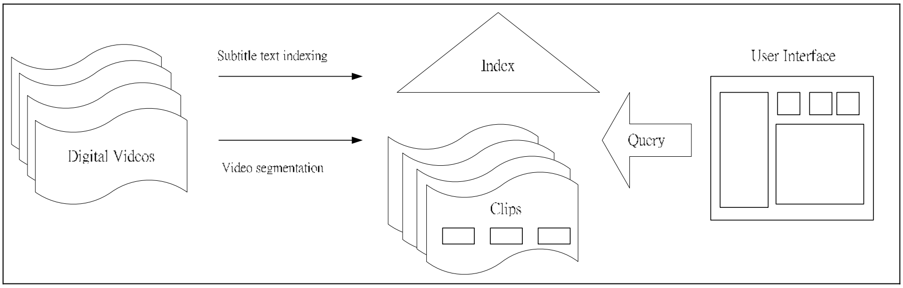
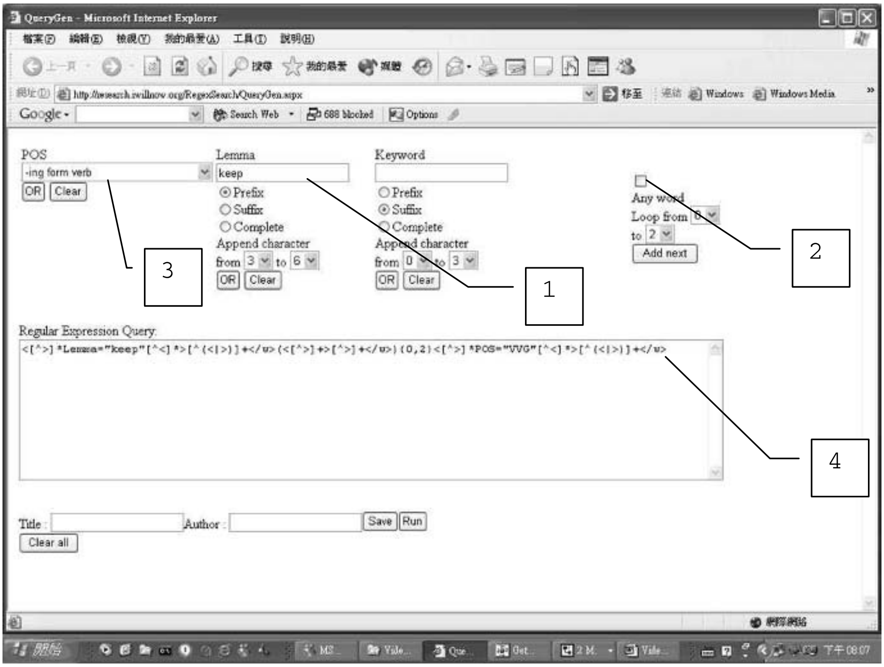


Figure 1: The architecture of the design video retrieval system

The architecture of the design video retrieval system is shown in Figure 1. As we mentioned in the previous section, the system provides a complete scene as the result of user input. Therefore we need to segment the video into clips. The video segmentation approaches will be described in Section 3.2. Meanwhile, for standardized query and fast retrieval, the movie subtitle has to be well-formatted and indexing. As for subtitle text preprocessing, in order to provide syntax query, these text are all part-of-speech tagged. We design a Markov Model-based POS tagger [4] and use British National Corpus[1] (BNC) as our training data. The internal evaluation shows this tagger has 93% precision including identifying unknown words. After preprocessing, these texts are standardized to predefined XML format for regex search.

Query processing system is in charge of matching the query terms and index terms, and then replying to the search result. However, by using regular expression as our query language, this background knowledge of regex would become a bottleneck for our target users. Thus, we design a syntax regex query generator shown in Figure 2. This query interface bridges the gap between users and regex, i.e. the users do not have to learn anything about regex.


QueryGen - Microsoft Internet Explorer
http://research.iwillnow.org/RegexSearch/QueryGen.aspx
Google
Search Web
688 blocked
Options
1
2
3
4

POS: -ing form verb

OR Clear

Lemma: keep

● Prefix
○ Suffix
○ Complete

Append character from 3 to 6

OR Clear

Keyword:

○ Prefix
● Suffix
○ Complete

Append character from 0 to 3

OR Clear

□ Any word

Loop from 0 to 2

Add next

Regular Expression Query:

```
<[^>]*Lemma="keep"[^<]*>[^(<|>)]+</w>(<[^>]+>[^>]+</w>){0,2}<[^>]*POS="VVG"[^<]*>[^(<|>)]+</w>
```

Title: Author: Save Run

Clear all

Figure 2: The designed User Interface of Query Generator,

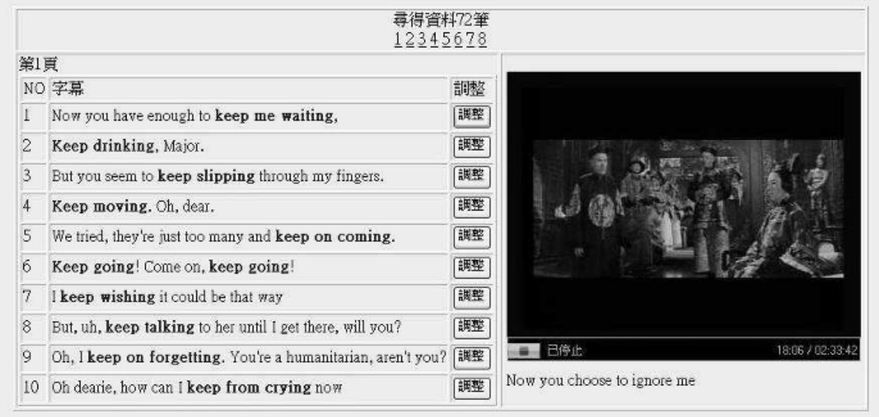
We will use the following example query to show the detailed flow from inputting query to final result representation in this section. Here is the example:

A teacher wants students to learn "verb keep has to be followed by -ing form verb".

The teacher can use the query generator to produce the regex query, shown in "Regular Expression Query" textbox of Figure 2. The query is composed by simple linear syntax knowledge. As this example, first we input *keep* as first query term (legend 1 in Figure 2). Since the -ing word can be several words right from *keep*, we can insert couple words for this elasticity of demand (legend 2 in Figure 2). The last query term, of course, is the part-of-speech "-ing form verb" (legend 3 in Figure 2). The corresponding regex query is generated as legend 4 in Figure 2. From several personal contacts with English teachers in Taiwan, this linear syntax search engine would be beneficial to many English teaching conditions.

The Query Generator then sends the query to the query processor. There are two kinds of data sent to the query processor. One of them is regex query, which would be used in the final step of retrieving process. The other is the information of query terms. These query terms are used to narrow down the search target, which can improve the response time. Even though regular expressions provide more flexible querying, they still create a serious problem dealing with slow search times. For example, a text collection with 1 million documents and 1000 words average length would take an unacceptable response time, say, a couple hours, by match the above sample query pattern to strings of text. Without further processing, the only way to find the pattern is to scan each document one by one in the text collection. The index construction which is used to solve this problem will be described in Section 3.1, and here we only assume that the subtitle text have been already well-indexed for matching the query terms. In this example, the query terms would be "*keep*" and "VVG", which is the part-of-speech tag of "-ing form verb".

Using these two query terms, the system can provide a candidate set of clips which subtitle text contain both these two query terms. Although these candidates contain the two query terms, it does not mean that they are the final results. Maybe these two terms are very far away from each other in subtitle text. So they still have to be identified by regex query. Figure 3 shows the final results of this example. This figure is the integrated environment of search results and IWiLL authoring environment, which we will mention in Section 4.



尋得資料72筆
1 2 3 4 5 6 7 8

第1頁

NO	字幕	調整
1	Now you have enough to **keep me waiting**,	調整
2	**Keep drinking**, Major.	調整
3	But you seem to **keep slipping** through my fingers.	調整
4	**Keep moving**. Oh, dear.	調整
5	We tried, they're just too many and **keep on coming**.	調整
6	**Keep going**! Come on, **keep going**!	調整
7	I **keep wishing** it could be that way	調整
8	But, uh, **keep talking** to her until I get there, will you?	調整
9	Oh, I **keep on forgetting**. You're a humanitarian, aren't you?	調整
10	Oh dearie, how can I **keep from crying** now	調整

Now you choose to ignore me

Figure 3: Search results of the corresponding example *keep* + *Ving* form

3. Methodologies

3.1 Syntax search engine

In this section, we will show how the syntax search engine is implemented. The basic matching scheme is to use regular expressions. However, as mentioned in Section 2, regex matching by scanning entire database takes too much time. So the situation is how the whole retrieving process can be sped up. The easiest way to solve this problem is to reduce the scanning data size. Due to the usage of regex, we applied k-gram indexing to achieve this goal, which is based on [5]. As for query processing, it can be referred to [6].

First we introduce some notations and definitions. A k-gram is a string $x = x_1 x_2 \cdots x_k$, where $x_i : 1 \le i \le k$ is a character. A data unit means the unit in which the raw data is partitioned, such as web page, a paragraph or a sentence in documents. Let *M(x)* denote the number of data units which contain *x* and then the filter factor is denoted by $ff(x) = 1 - M(x)/N$, where *N* means the total number of data units. Now we can set the minimum filter factor *minff* and only keep index terms with filter factors greater than *minff.* This would make the number of index terms smaller and more useful. For example, *minff* = 0.95 means the system only keeps the index term which can filter 95% of data units. This filter scheme can make the index terms more discriminative, which is important for a retrieval system. We call these index terms **useful** indices.

Even if the system only maintains useful indices, the number of indices is still quite large. Every string expanded from a useful index will be useful, too. For example, if the index **NBA** is useful with in the text "**How to buy NBA tickets**", then "**y NBA**" and "**NBA t**" are useful but not necessary. Therefore, the system only maintains a presuf (prefix and suffix) free index set. A presuf free set means there is no *x* in the index set is a prefix or suffix of any other index x'. For example, {*ab, ac, abc*} and {*bc, ac, abc*} are not presuf free because *ab* is a prefix of *abc* in first set and *bc* is a suffix of *abc* in second set. The complete index construction algorithm is as follow.

```
Input：text collection, minK, maxK
Output：index

k=minK，Useless={.} // . is a zero-length string
while (Useless is not empty) & (k<=maxK)
    k-grams：= all k-grams in text collection
      whose (k-1)-prefix∈ Useless
      or (k-1)-suffix∈ Useless
    Useless：={}
    For each x in k-grams
        If ff(x) ≥ minff Then
            insert(x,index)  //the gram is useful
        Else
            Useless：=Useless ∪ {x}
    k：=k+1
```

After determining index terms, we can construct the index for a regex search engine. We use inverted indices as our index storage structure, which is easily accessed by RDBMS. What is different in our steps from the original one [5] is we do prefix checking and suffix checking in the same pass. We need to scan entire data once for extracting all k-grams. After the index is built, it still needs one more entire data scanning for identifying index term positions. Therefore, the whole index construction only needs two entire data scannings without any extra memory space. However, if the memory is large enough, we can extract k-grams and get the positions in the same data scanning pass.

As mentioned before, the k-grams index is used to reduce the number of data units to be matched by regex. For this reason, the query processor has to determine which index term to look up. Like the example in Section 2, *keep* might be a useful index but filtered by presuf-free process. However, assuming *k*=3-10,and *kee* and *eep* are all index terms, the system then can reduce the size of the target regex matching data. As for how to choose the right index term (*kee* or *eep* or both), the strategy can be referred to [6], which also mentions the significant performance.

3.2 Video Segmentation

The other major design in this system is how to segment digital video into clips. As mentioned in Section 1, our purpose in designing this tool is to provide a whole scene for lighten the burden of the authoring job. We applied the segmentation algorithm from [7]. This algorithm assumes that different scene clips will have different color space distribution. By computing the similarity of each image extracted from videos, this algorithm can detect scene change points in each video. Every pixel in the image has three values (R, G, B) to represent the color space. We use color histogram to describe the color distribution of one image. The similarity between two images then can be computed by:

$$\textit{distance}\ (H_A, H_B)$$

where H_A and H_B are the vectors of color histogram for Image A and B. By setting a threshold ε, we can detect if there exists a scene change point between two sequential images. While the distance greater thanε, it means that the two images are dissimilar, i.e. there might exist a scene change. We show an example of our implementation result in Figure 4.

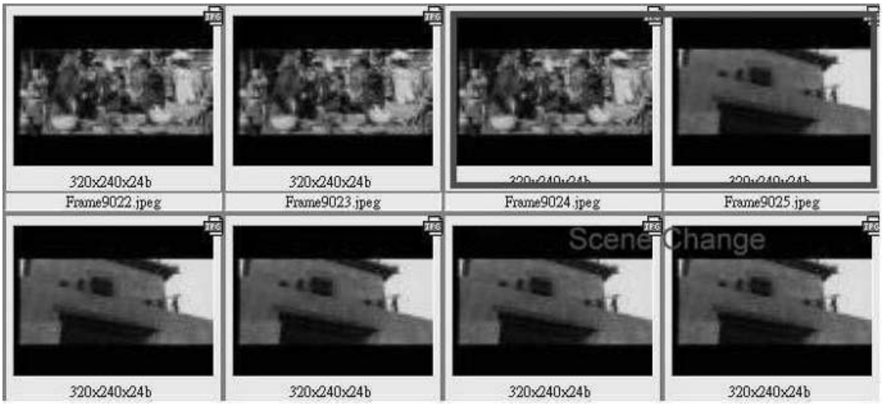


Figure 4: The implementation result of scene change detection

4. Integration of video retrieval system in the IWiLL system

The IWiLL system [3, 8] consists of many language tools, e.g., collcation explorer, syntax-based retrieval system, and interactive learning environemnt, e.g., discussion board. These elements can be integrated together by using the designed authoring tool. The fundamental design philosophy of the authoring tool is shown in Figure 5. The designed content can be shared among the users of the IWiLL system, including videos. As an example, we illustrate the designed content in Figure 6. The designed authoring tool is of ease to use.

English teachers with common IT knowledge are able to manipulate this tool after short time training.

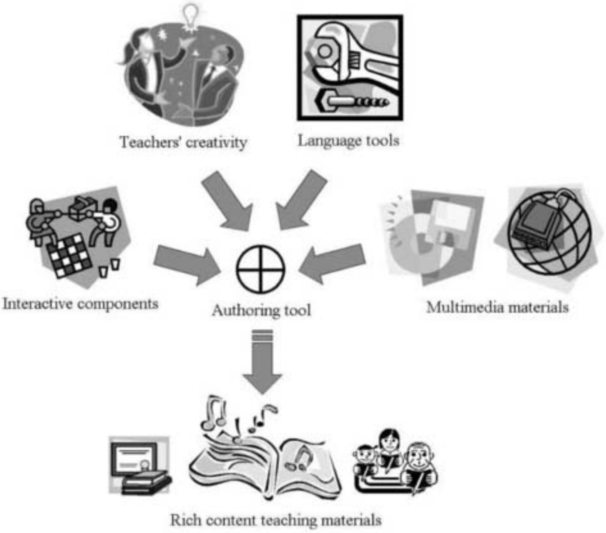
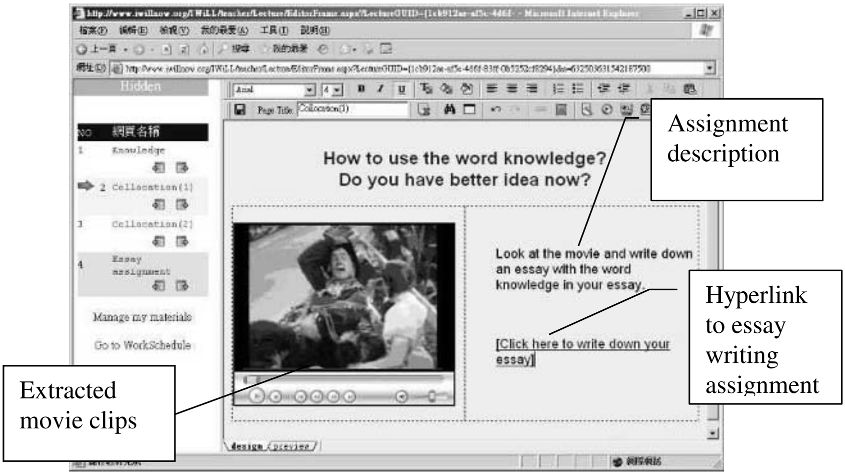


Figure 5. The authoring tool: design philosophy


How to use the word knowledge? Do you have better idea now?

Look at the movie and write down an essay with the word knowledge in your essay.

[Click here to write down your essay]

Assignment description

Hyperlink to essay writing assignment

Extracted movie clips

Figure 6. Learning content example: Writing assignment

Authentic learning environment is beneficial to language learner, particularly, to second language learners. In the IWiLL system, we have built a video database to provide videos as learning materials to learners. Note that the purpose of the integration of these language tools and interactive components is to create a sprial learning effect to target language learning [3].

5. Conclusion

Currently there have been few managing tools for dealing large video database, especially for language learning purpose. In this paper we present a video retrieval system with syntax search ability. In order to achieve quick response and regular expression matching, a k-gram indexing algorithm is proposed. We designed automatic video segmentation for lighten the authoring burden. The integration of this video retrieval system and a teaching material authoring tool is also shown. However, using digital video for language learning still need teachers' creativity. We hope this system can allow teachers to fully utilize digital video with more time.

Notes

[1] http://www.natcorp.ox.ac.uk/

References

[1] Jane King, "*Using* DVD Feature Films in the EFL Classroom," *Computer Assisted Language Learning*, Vol. 15, No. 5, pp 509-523, 2002.

[2] Erwin Tschirner, "Language Acquisition in the Classroom: The Role of Digital Video," *Computer Assisted Language Learning*, Vol. 14, No. 3-4, pp 305-319, 2001.

[3] Chin-Hwa Kuo, David Wible, Tzu-Chuan Chou and Nai-Lung Tsao, "On the Design of Web-based Interactive *Multimedia* Contents for English Learning," in *Proceedings of IEEE International Conference on Advanced Learning Technologies (ICALT)*, Joensuu, Finland, August 30- September 1, 2004.

[4] Thorsten.Brants, "TnT-A Statistical Part-of-Speech Tagger," in *Proceedings of the Sixth Applied Natrual Language Processing Conference ANLP-2000*, Seatle, WA, 2000.

[5] Junghoo Cho and Sridhar Rajagopalan, "A Fast Regular Expression Indexing Engine," in *Proceedings of 18th IEEE* Conference *on Data Engineering*, 2002.

[6] Nai-Lung Tsao, Chin-Hwa Kuo and Meng-Chang Chen, "Designing a Parallel Regular-Expression Search Engine", submitted to *ACM Fourteenth Conference on Information and Knowledge Management (CIKM)*, 2005.

[7] Alberto Del Bimbo, Visual Information Retrieval, Morgan Kaufmann, 1999.

[8] David Wible, Chin-Hwa Kuo, and Nai-Lung Tsao, "Contextualizing Language Learning in the Digital Wild: Tools and a Framework," *IEEE International Conference on Advanced Learning Technologies (ICALT)*, Joensuu, Finland, August 30- September 1, 2004.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

The Activity at the Center of the Global Open and Distance Learning Process

Lahcen Oubahssi*, Monique Grandbastien**, Macaire Ngomo***, Gérard Claës***

** Université René Descartes Laboratoire CRIP5 /AIDA 45 Rue Saints Pères*
75270 PARIS Cedex 06
oubahssilahcen@voila.fr
*** Université Henri Poincaré Nancy1 Lab. LORIA/AIDA – Bât. LORIA Campus scientifique*
BP 239 - 54506 VANDOEUVRE Cedex France
monique.grandbastien@loria.fr
**** Société A6, 42 rue Paul Claudel 91000 EVRY France,*
macaire.ngomo@wanadoo.fr gerard.claes@wanadoo.fr

ABSTRACT: Our global objective is to propose models and functional architectures for the open and distance learning (ODL) systems that are elaborated from the practices observed within a company marketing ODL platform-based solutions. Therefore it is a re-engineering process whose characteristic feature is to embrace the overall open and distance learning life cycle.
In this paper, we focus on the concept of activity. A lot of propositions are centered on the learner's activity. First we describe how the concept of activity is used in some representative models in existing literature, then we propose a more extensive model of activity that covers all the actor's activities involved in the open and distance learning production process. Finally, we show how this model is used in several situations.

KEYWORDS *:* Model, Activity, Process, EML, IMS LD, Open Distance Learning Production.

1. Introduction

The first Open and Distance Learning platforms made it possible to provide the learners with learning contents and various communication functionalities with other learners or their teachers. Most on-line learning still functions in this way. However, training cannot be reduced to a simple transfer of knowledge through the provision of resources. The acquisition of knowledge and know-how comes from many sources and results from varied activities [1] like, for example, solving problems, interacting with genuine tools and collaborating with other actors. To improve and diversify distance learning, it thus appeared necessary to study the activities within the training process, particularly the learner's activites, then to describe and organize these activities.
Describing the activities, in relation to the resources and the actors, requires frameworks or specification languages adapted to these needs and largely accepted to allow exchanges and reutilisability. Hence, educational modelling languages appeared and various specifications were also proposed to standardize the exchanges in the on-line training domain. We have classified these propositions into three categories: those which model the resources (ARIADNE[1], CanCore[2], Dublincore[3], LOM[4]…), those which model the activity in particular

[1] http ://www.ariadne-eu.org/
[2] http ://www.cancore.ca/fr/
[3] http://dublincore.org/
[4] http ://www.afnor.fr
[5] http :// eml.ou.nl/eml-ou-nl.htm
[6] http://www.imsglobal.org/learningdesign/
[7] http ://www.imsglobal.org/profils/lipbest01.html
[8] http ://www.imsglobal.org

the pedagogic activity (EML[5], IMS LD[6],…) and those which model other elements such as the learner's competences (LIP[7], IMS RDCEO[8],…). In this paper, we are interested in the modelling of the activity. An activity can be defined as a set of actions that transform resources into results. It is performed in an ODL environment by one or several actors who use tools and services offered by the environment.

Most propositions are centered on the learner's activity. Our objective is to start from a more global activity model within a global ODL process and to adapt it to each process phase. We should thus be able to ensure a better interoperability of the data pertaining to these activities between the various software components used in an ODL. Before describing the concept of activity in some existing models, we present a partial view of the activity in the general ISO production process, and in particular in the global ODL process. We then propose an activity model that covers the activity of all the ODL process actors more largely. We finally show how this model is used in several situations.

For ODL systems, our approach is to propose models and functional architectures resulting from the practices observed within a company which markets solutions around a learning management system. Therefore it is a process of re-engineering whose characteristic feature is to cover the set of the ODL life cycle.

2. Global ODL Process

Many models have already been proposed for open and distance learning and recently for the delivery of on-line learning. Most models take a partial view of the activity or concentrate on a given category of actors. Our goal is to build an activity model that considers the whole life cycle of the open and distance learning production.To this end, we start with some general models reflecting the industrial production process.

2.1. The ISO production process model

The ISO 9000[9] standard assesses the quality of a learning process. At the level of ISO 9000, a process is defined as a set of interactive and interrelated activities, which transform input elements into output elements. The input elements of a process generally constitute the output elements of one or several other processes.

Several processes may occur in a product life cycle, and they describe the means and activities, that transform the input data into output data. A process itself is composed of a set of transformations which adds value to the input data. These transformations depend and rely on external factors and resources, namely performances, material and human resources.

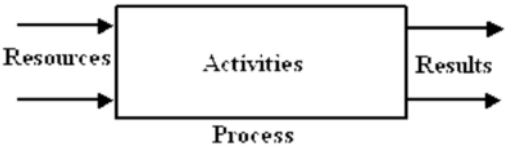


Figure 1 : The ISO production process model

The **resources** represent the process input data

The **results** represent the process output data

The **activity** represents an act which allows the transformation of the resources into results and as seen in Figure 1 the activity constitutes a central element of the process.

2.2. Open and Distance Learning model:

From these definitions we can derive a process-oriented view of on-line learning production. For open and distance learning production, input data include knowledge, know-how, and

[9] http ://www.iso.org

curricula. Data suppliers are teachers, trainers, designers of training resources, technicians, administrators and specialists in other domains. Output data include training sessions, evaluation and testing modules, scores and additional information about the learner. The main customers for these data are the learners. The global process is made possible by external factors such as material resources (equipment, computer-based services) and human resources (teachers, tutors, training and administrative staff). Other constraints or success criteria are described under the performance items (financial cost, quality management, and success criteria) and the progress (duration, and calendar constraints).

From the industrial point of view, it is very important to start from a process -oriented approach that considers the producing of a training activity exactly like any other production process within a company. However, we need complementary views to focus our attention on the way sub-processes are scheduled and on the support these sub-processes are given or not by existing services.

The complete ODL life cycle follows five principal phases in [2] : creation phase, orientation phase, training phase, follow-up and evaluation phase and management phase. Each phase calls upon the succession of several processes- in their centre, increasingly detailed activities are found. The process activities are perfomed in environments related to each phase of the complete ODL life cycle.

In the next paragraph we present some activity models, in particular the learner activity. The selected models are the following: EML, IMS LD. And then we present our model of activity in the ODL process.

3. Analyzing few existing models

3.1. EML model (Koper)

The initial EML (Educational Modelling Language) came from the work completed in Open University of the Netherlands on the design and the development of a description language adapted to education. The work started in 1998 aimed at creating a notation equipped with a semantics to describe training situations.

In his model, Koper [1] proposes to describe the effective training situations using a Educational Modelling Language which places the training situations and not the resources in the center of the process. We should bear in mind that the first proposals resulting from different consortia (e.g. ARIADNE, IEEE/LTSC[10]) were only related to the description of resources.

The main concept in the EML model is that of unit of study [3]. Typically, a unit of study can be a course, a lesson, a case study, a practical work.

The study unit, must answer the following constraints :

- A study unit corresponds to a precise teaching objective and requires a certain number of prerequisites.
- A study unit is made up of a set of activities.
- An activity is carried out by one or more actors having their specific role.
- The actor can be a learner or a staff.

Each activity, characterized by a set of prequisites and pedagogic objectives, is defined by a state (for example, finished).

The concept of environment in which the activity proceeds makes it possible to gather a set of resources of any type. Thus the EML model defines the following types of objects : knowledge objects, communication objects, tools objects and test objects. EML model also defines other classes of objects making it possible to manage the structuring of the activities,

[10] http://ltsc.ieee.org

the roles and the resources such as the property objects, the section objects, the index objects, the research objects, the advertisement objects, etc.
In the EML model, the study unit is viewed as a composition of activities carried out by actors in a given environment; activities can be distinguished among following : the learning activity, the support activity and the instrumental activity.

3.2. IMS LD Model

Other EMLs exist. A first synthetic work on EML [4] was carried out by the CEN/ISSS working group on training technologies. The results were used within the working group "Learning Design" of the IMS consortium which in November 2002, brought up a proposal for a specification designed to become a standard, IMS-Learning Design.
The IMS-Learning Design model rests on the following principles :

- A person holds a role and achieves activities by possibly using resources, services and tools.
- Each person can have one or more recordings which have properties characterizing it.
- There are two generic records : "staffs" and "learner".

In IMS LD, the activities characterized by objectives and prerequisites have a specific structure, use resources and produce results.These results can be injected again into other activities.

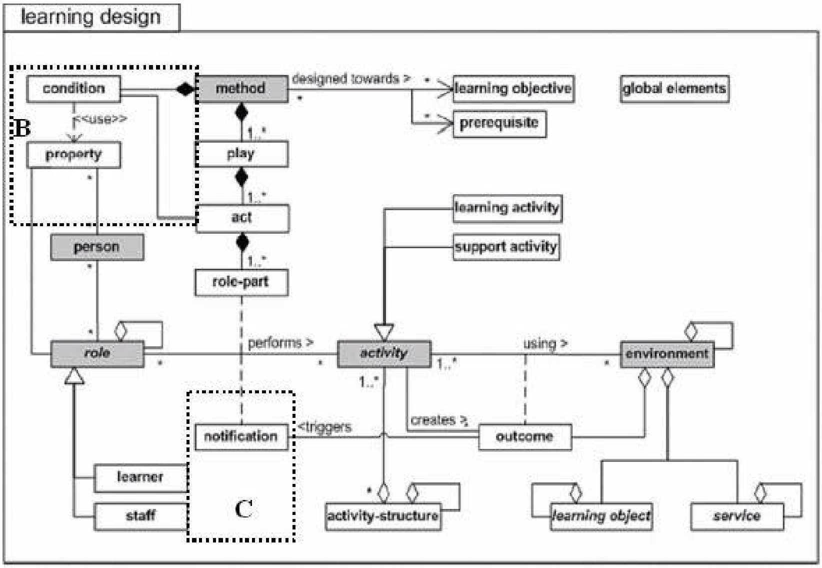


Figure2 : IMS LD Model

The IMS-LD model allows to specify the progress of a training unit, it uses the LOM for the metadata description relating to the resources and recognizes the pedagogic objects as part of the learning environments. It also places the activity at the center of the process, figure 2 shows the relations between the various concepts selected.
The "learning design" can be made at three levels :
On the first level, one can conceive only predictive scenarios without taking into account, the learning results in the activities sequence.
On the second level one can design a learning model and take into account, in the activities sequence, one can individualize the course of the scenario.

The third level offers a simple means to synchronize the multiple processes which take place during a training unit.

The IMS LD model is very close to the EML model from which it results, but presents some differences : in the place of the study unit, it uses the concept of the training unit, it also uses the resource concept instead of the object, and finally, an activity not only can use resources, but can also produce new ones. Like in EML, we can note that the global ODL life cycle is not totally covered.

3.3. Conclusion

There are other models describing from different perspectives the activity, some inspired by the models presented above.

J.-P. Pernin [5] proposes a conceptual model based on the concept of pedagogic scenario. His proposal rests on a set of well defined concepts and on a taxonomy of scenarios. This model includes the activities and focuses on the type of relations binding activities and resources.

G. Paquette [6] proposes a complete method of pedagogic engineering MISA, which covers the design of the requirements until the implantation within the Explor@ platform. The concepts of knowledge and competences constitute the groundwork of modelling. As in IMS LD, the activities intervene in the training units description.

We can note that most of models presented take a particular view of the activity or concentrate on a given actors's category. In the following paragraph, our goal is to propose a model which takes into account all the activities of the Open and Distance Learning process life cycle.

4. An activity model for global Open and Distance Learning process

4.1. Activities in the global Open and Distance Learning cycle

Before detailing our proposal, we can briefly recall, the principal activities which this model will have to take into account. We have grouped these activities according to the five phases of the global ODL cycle [7].

During the creation phase, the author uses his creation environment to carry out the following activities : design and development of the pedagogic modules, preparation and integration of the pedagogic modules contents, test and simulation of the pedagogic modules, diffusion of the validated modules, collaboration and cooperation with the other actors,etc...

During the orientation phase, the adviser uses his orientation environment to carry out the following activities : elaboration of the training plans, elaboration of the learner’s curriculum, development of the learning courses for the groups, development of the learner’ booklets, development of plannings, collaboration and cooperation with the other actors,etc...

During the learning phase, we can distinguish two principal actors, the tutor and the learner. This later follows his learning sessions (by having access to the pedagogic modules and carrying out assessment tests), he collaborates with his group’s members and his tutors. As for the tutor, he leads the learning sessions and he analyzes the sessions feedback.

During the evaluation phase, the examiner uses her evaluation environment to carry out the following activities : development of the evaluations, development of the learning follow-ups, collaboration and cooperation with the other actors,etc...

During the management phase, each manager and administrator use their environment to accomplish the following activities :

- activities of administrative management : handling of the users accounts (learner and teacher), the group accounts, the schedules and the administrative agreements ...

- activities linked to technical management: securisation of the data, maintenance of the pedagogic course, management of the documents...
- activities of learning management : defining the learning fields, defining of the disciplines, defining the training levels, and handling the documents.

4.2. A global model of the activity

Our model is designed to include the IMS LD model whose only objective is to describe a training unit. We describe an ODL environment, made up of particular working units which can be training units. Figure 4 provides a class diagram which details the model of the activity proposed in ODL environment. An environment is associated to each phase of the process in which the actors carry out one or several activities. In this model, an ODL environment is thus composed of a set of working units, rules and resources.

Activities are held within the working units. The working unit is defined as a composition of activities carried out by a set of actors in a given ODL environment. We can distinguish five types of working units : the creation unit, the orientation unit, the training unit , the evaluation unit and the management unit. Each activity is characterized by its prerequisites and its objectives, and is defined by a state (for example, in progress). The environment in which the activity is perfomed makes it possible to collect the resources and the tools necessary to carry out activity.

Each activity uses and produces a set of resources (tools, services, results...). The principal actors who handle the activities are the following : the author, the adviser, the tutor, the learner, the evaluator, the staff, the general administrator, and the teaching administrator.

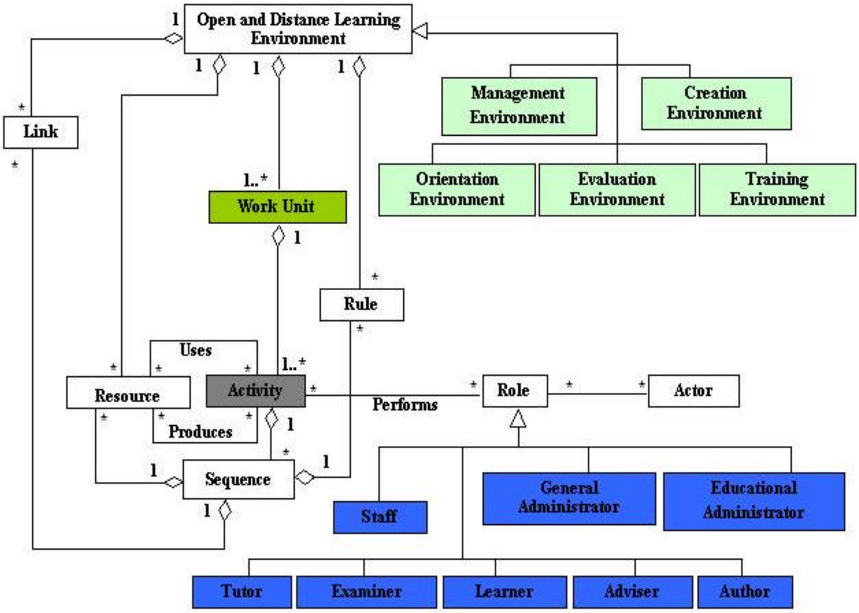


***Figure 4 :** A global model of the activity*

The rules represent the conditions or the constraints which allow the good progress of the activities.

In this model we tried to present a comprehensive view of the activity in the ODL process, that led us to define a new concept "working unit". This concept makes it possible to distinguish the activities along the five phases of the process.

In the following paragraph, we detail two examples of possible uses of our model, first in orientation activity, second in a training activity.

4.3. Use of this model

4.3.1. Example 1: Orientation activity model

Figure 5 represents the class diagram of the activities performed in the orientation phase of the cycle. Indeed, the orientation environment is one of the ODL environments, it consists of a set of rules, of links, resources and orientations units, which distinguishes it from the other environments. An orientation unit is considered as a composition of orientation activities carried out by the counsellors. We can distinguish several activities among orientation eg : the developing training plan, learner's curriculum, and exploring learner's follow-up...

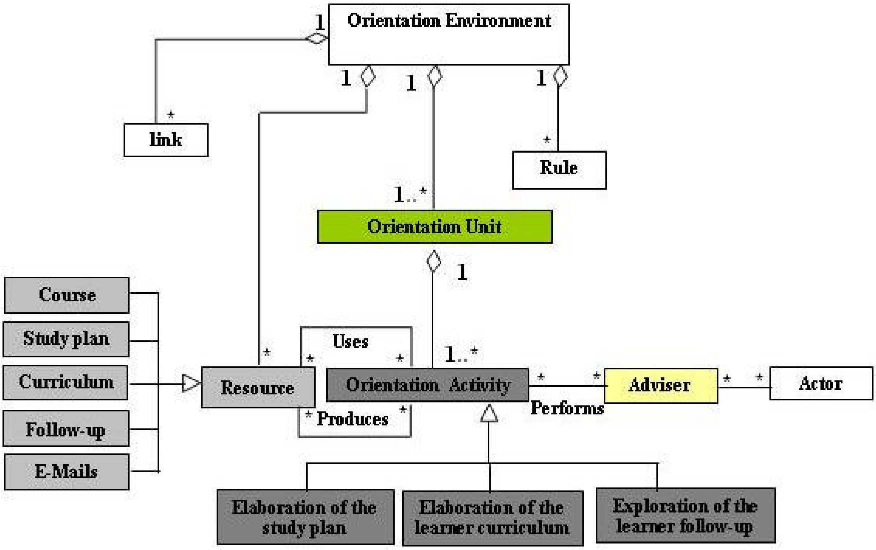


Figure 5 : Diagram of the orientation activity

The orientation activities use resources produced by the creation activities in creation units such as the pedagogic modules. They produce resources such as the learner's curriculum which will be used in the teaching activities. Other examples of resources used or produced by the orientation activities are the study plans, the learner's follow-up...

4.3.2. Example 2: Pedagogic activity model

The pedagogic activity lies at the center of the ODL process. Indeed, the learning phase and, in particular, its pedagogic activities relate to the most significant entity of the ODL process, learning. Consequently the models presented in the beginning of this document deal with this type of activity more particularly.

Figure 6 represents the class diagram of the activities carried in the training phase of the ODL cycle. In this phase, the work unit becomes a training unit, in which the tutors and learner carry out their activities. Using resources and producing others are possible. The resources used and produced by these activities are :

- tools and services such as : chat, forum, Emails, these tools are used by the learner and the tutors in a communication or collaboration context.
- pedagogic modules, courses, follow-ups, and raw documents.

The training units of IMS LD can be found in such a model.

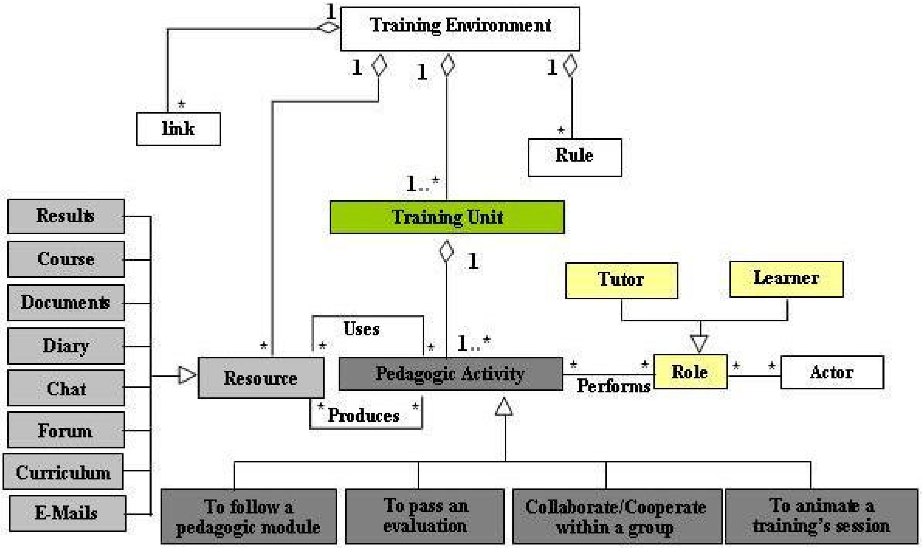


Figure6 : Diagram of the pedagogic activity

Conclusion

If we try to compare our model to the other models, we can note that it carries out an extension which makes it possible to apprehend the activities of the various actors who intervene throughout the ODL life cycle. From the industrial point of view, it is essential to have models making it possible to establish not only the training units themselves but also the management of the set of devices (e.g. associated resources and rights, authors, trainers and competences, tutors and payments). Next step is the implementation of such of model in order to get feedback from end users.

This proposal is currently supplemented by an accurate study of the data exchanges related to the resources used or produced by the activities at each phase of the process, partially presented in [7]. These global models also aim to rather establish certain functions through accessible on-line services than attach them in a formal way to a platform configuration.

Bibliography

[1] R.Koper. «Modelling units of study from a pedagogical perspective the pedagogical meta-model behind EML». Educational Technology Expertise Centre. Open University of the Netherlands. http://eml.ou.nl/introduction/docs/ped-metamodel.pdf

[2] M.Grandbastien, L.Oubahssi, G.Claës. «A process oriented approach for modelling on line Learning Environments», *in Intelligent Management Systems, AIED2003 supplemental proceedings,* vol.4, pp. 140-152., university of Sydney pub., 2003.

[3] R.Koper. «Combining re-usable learning, resources and services to pedagogical purposeful units of learning». *In A. Littlejohn (Ed.), Reusing Online Resources: A Sustainable Approach to eLearning* (pp. 46-59). Kogan Page, London 2003. ISBN 0-7494-3950-5.

[4] A. Rawlings, P. Van Rosmalen, R. Koper, M. Rodriguez-Artacho, P. Lefrere. «Survey of Educational Modelling Languages (EMLs), version 1». September 19th 2002, CEN/ISSS WS/LT.

[5] J-P.Pernin, A.Lejeune. «A taxonomy for scenario-based engineering». Cognition and Exploratory Learning in Digital Age (CELDA 2004) Proceedings, p.249-256, Lisboa, Portugal, dec. 2004.

[6] G.Paquette. «Instructional Engineering in Networked Environments». 304 pages. January 2004. Publisher: Pfeiffer & Company. ISBN: 0-7879-6466-2.

[7] L.Oubahssi, M.Grandbastien, G.Claës. «Ré-ingénierie d'une plate-forme fondée sur la modélisation d'un processus global de FOAD », *Colloque TICE2004,* pp. 32-38. Octobre 2004, Université de Technologie de Compiègne.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Towards support in building qualitative knowledge models

Vania BESSA MACHADO, Roland GROEN and Bert BREDEWEG
Human Computer Studies, Faculty of Science, University of Amsterdam,
Kruislaan 419 (matrix 1), 1098 VA Amsterdam, The Netherlands
Email: {vbessama, bredeweg}@science.uva.nl

Abstract. Qualitative Reasoning (QR) formalisms provide ontological primitives for capturing *conceptual* knowledge. Recently QR-based diagrammatic tools are being developed to support learners in creating concept maps as means to acquire such knowledge. However, QR formalisms are complex which hampers their usability. While other approaches address this by simplifying the formalism they use, we seek the solution in providing a set of agents that can support the learner. Based on a previously reported study on using QR modelling tools, we have developed a multi-agent approach to support the QR modelling process. The agents provide different kinds of help, such as general infomation on the formalisms and tailored feedback addressing the individual needs of a learner. Agents thus have scope, provide context-sensitive help, and are personified according to the type of support they provide. An evaluation study shows that the help-system is well accepted by learners.

1. Introduction

Conceptual analysis of systems and their behaviour is a central skill in scientific reasoning. Enabling and encouraging the creation of domain theories, which can be instantiated to specific situations, helps learners to understand the broad applicability of scientific principles and processes. The research area Qualitative Reasoning (QR) provides means that can aid this kind of learning. QR formalisms provide a way to express *conceptual* knowledge such as system structure, causality, the start and finish of processes, the assumptions and conditions under which facts are true, qualitative distinct behaviours, etc. Models provide formal means to externalise thought on such conceptual notions. Particularly the idea of having learners learn by *building* qualitative knowledge models enables them to formulate their own ideas, test them by simulation, and revise them were needed [6, 9]. These are important scientific skills for learners to acquire.

QR formalisms are complex and therefore not always easy to use in educational settings. Recently tools are being developed that take a graphical approach to having learners build qualitative models [5]. Graphical representations help reduce working memory load, allowing students to work through more complex problems. Such external representations also help them present their ideas to others for discussion and collaboration. This closely relates to the idea of using concept maps [8]. The main difference being the rich and detailed semantics used, based on QR formalisms. To further enhance usability, approaches such as Betty's Brain [3] and Vmodel [7] reduce the amount of primitives available in the model-building tool. Although this is effective, it has the obvious drawback of not using the full potential of QR and the means it provides to articulate conceptual knowledge. In our approach we want to preserve the full expressiveness of the QR formalism. To enable usability we have develop support tools that aid learners in understanding the representational primitives (which we regard as an important learning goal by itself) and to articulate and reflect on their thoughts.

This paper discusses the multi-agent help system that we have developed for the domain-independent model-building environment MOBUM [1]. It builds on previous work [1, 2] in which we used the workbench Homer to evaluate the usability of a diagrammatic representation for qualitative knowledge and the need for additional help, both from a learner perspective. The evaluation of Homer was designed such that we obtained as much information as possible on problems that learners encountered when working with Homer.

Based on the insights gained from this evaluation MOBUM was constructed. MOBUM uses a related but improved diagrammatic presentation, compared to Homer. To further enhance usability, MOBUM was given a multi-agent help system that is capable of providing useful help without maintaining an explicit learner model nor a norm model.

2. MOBUM – a brief Overview

MOBUM is workbench for creating and simulating qualitative knowledge models. It is based on the QR formalism described in [4]. The graphical user interface of MOBUM is organised as a set of builders and tools. Builders are interactive windows that support the learner in building specific model ingredients. In the current version of MOBUM there are five builders that support the creation of these model ingredients, namely for: Model fragments, Quantities, Quantity spaces, Entities and Scenarios. Two others builders exist that do not directly add content to the model, but support the learner in exercising his/her understanding of the system being modelled. These addition builders provide means for expressing ideas using drawings (SWAN SketchPad) and causal dependencies (Causal Model Builder). In addition to the builders there is a set of Model Inspection Tools, which allow the learner to run a simulation, to visualise the global simulation results (e.g., state-graph) and to inspect the specific results of the simulation (e.g. the contents of an applied model fragment). Thus, after running a simulation, the modeller will get a state-graph and can verify, for instance, how the quantities behaved in the different states, which model fragments have applied, the content of a specific state, and how the transition occurred from one state to another.

The diagrammatic representation of model ingredients within the builders follows the guidelines presented in [2]. For example, *Quantities* in the Quantity Builder are organised in a list, because no relation exist between them, while *Entities* are represented as nodes in a graph and the *is*-a relation between the entities are represented as arcs between those nodes. An example of what a learner may produce is shown in Figure 1.

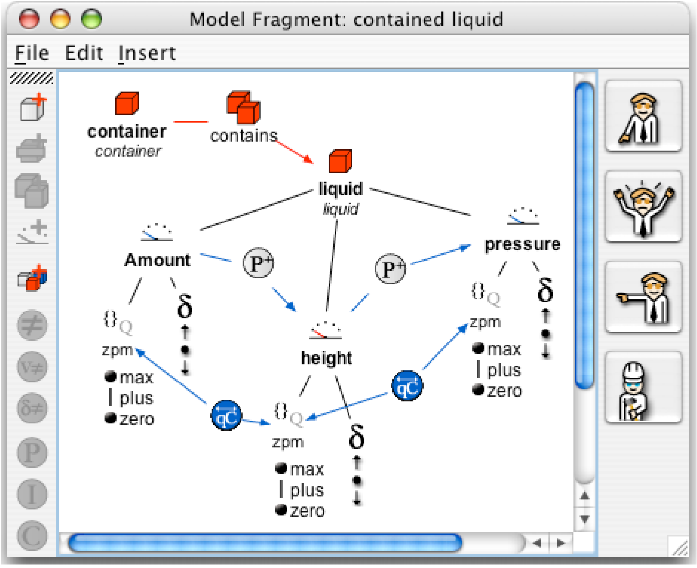


Figure 1: Model fragment of a ‘Contained Liquid’

The figure shows the Model Fragment Builder that captures generic knowledge about a container containing a liquid, hence ‘contained-liquid’. The single cube-like icons represent objects (*container* and *liquid*). The double cubes represent structural relations between objects (*contains*). There are three quantities: *amount*, *height*, and *pressure*. They are assigned to the *liquid*. Each of these quantities can take on three possible values *{zero, plus, max}* and they can be *increasing*, *steady* or *decreasing* (∂) (although in this example no specific values nor derivatives have been assigned). *Amount* has a positive influence (P+) on *height*, and *height* on *pressure*, which means that when *amount* increases, so will *height* and *pressure*. These proportionalities (P+) are *directed* causal dependencies. Thus: a change in the *amount* causes the *height* to change and not the other way around. Finally, the quantity spaces for these three quantities fully correspond (qC), which means that they will always have the same value, e.g. all having value *max*. Notice that most of these model ingredients have been created with the other builders, such the Entities, Quantity, and Quantity space Builder. In the Model Fragment Builder these ingredients are re-used are related. In fact, only the Correspondences (qC) and the Proportionalities (P+) are actually newly defined in the Model Fragment Builder.

2 The Agent-based Help System

The design of the help system is based on the results from the study with HOMER [2]. The help system should be usable for a wide range of learners, active in different kinds of science teaching curricula. It should provide support related to conceptual knowledge, including the model-building ontology, and it also should provide tailored feedback addressing the individual needs of a learner.

Taking a domain independent approach has at least two consequences. Firstly, besides providing support to the learners in acquiring conceptual knowledge, support concerning the graphical language must also be given. As a result of being domain-independent, the icons used in MOBUM are generic and learners will most likely not immediately associate the underlying concepts with their visual representations. Secondly, the use of a learner model, in the traditional sense, is not possible because it would require a domain specific norm model to work from. To cope with this situation, we take a rather different approach compared to traditional ITS systems. Instead of focussing on the domain knowledge that the learner is supposed to acquire, we focus on the processes that are expected to lead to the acquisition of that knowledge. That is, we provide tailored feedback based on knowledge about the model-building process in general and the constraints following from the specific model built by a learner. Another feature of our approach is that the support system takes the form of an advisory system. We do not want to interrupt the learner in order to offer help. The learner is in control and can initiate a support session if needed.

Using pedagogical agents is a relative new paradigm. We assume that searching for help is more efficient when the support system is based on modular processes. We thus opted for an agent-based approach in which each agent is specialised in some specific task and together with the other agents collectively contributes to the achievement of a global objective. Agents, thus have scope, provide context-sensitive help, and are personified according to the type of support they provide. Two main categories of support were defined: static (pre-defined), dynamic (tailored to learner activities).

2.1 Structure of the Help System

Since the applicability of *static* and *dynamic* information is clearly delimited, their availability should also be broken down into discrete stages. Similar to what was done in the work presented in [10] and in order to stimulate the use of help as well as to unambiguously characterise each type of knowledge support, six agents presented as different characters are used (Table 1).

Table 1: Agents in the MOBUM multi-agent help system

	What is?	How to create?	Curriculum planner	Global help	What can I do next?	Cross builder help
Standby						
Active						

Each agent has a specific appearance representing the type of support it can provide. Each builder, representing a particular step in the model-building task, possesses its own implementation of the various agents (e.g., the model fragment builder has four of these agents, see Figure 1). The whole set of agents is thus present at all times but the support provided will depend on the actual model-building context.

2.2 Static Help

Part of the *static help* is implemented as two complementary forms. Firstly, by providing definitions for the terms composing the model-building ontology. Secondly, by giving examples on how to use those terms. The static help system is thus able to answer questions such as '*What is an influence?*' and explain '*How to create?*' an influence using the available tools.

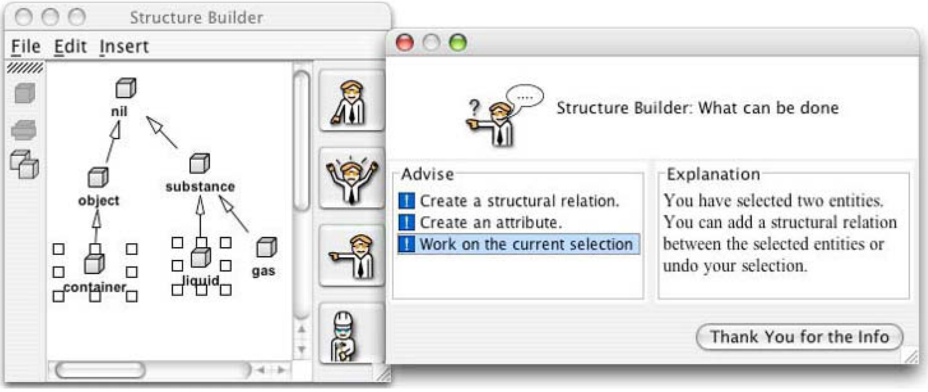
To support the learner in solving a problem, static agents use explanatory text, examples and images. The information is displayed inside a dialogue box using HTML pages including hyperlinks and cross-references. Images are also used for displaying MOBUM GUI parts. Four static agents are included in the design. They are labelled according to their specific utilities: *What is*, which has the task of helping learners on model-building concepts in the actual builder; *How to*, which suggests the order in which modelling steps should be performed and the actions needed to reach a certain goal; *Curriculum planner*, whose goal is to provide information related to specific assignments given to students; and *Global help*, which is knowledgeable about general modelling issues. It also explains the application of all ontological primitives and discusses basic ideas on how to create a model.

2.3 Dynamic Help

The dynamic help provides support relevant to the specific *content* of the model being created. This type of help thus needs to have *assessment* capabilities concerning the prior and actual user production in order to be able to evaluate the progress of the learners. Since this progression is a dynamic process, the contents of the provided help will be changing constantly. The dynamic help continuously analyses the current solution of the learner to the assigned problem and compares the steps taken to reach this point with a selection of generic correct modelling features. Any inconsistencies will be detected and can be reported to the learner so as to instigate the learner to reflect on the actions taken and maybe consider an alternative trajectory. By doing so we try to keep learners on track and to avoid them from arriving at incomplete models.

The dynamic help system is designed to provide guidance at two distinct levels: *local* and *global*. The former is concerned with the details of a specific modelling subtask and is usually restricted to a certain builder. The latter, on the other hand, gives a global perspective on the modelling activities of the learner, reuniting the actual status of the full model. This distinction between *local* and *global* knowledge is an important one, since the construction of models will usually be a constant interplay between figuring out the fundamental details of the underlying model ingredients and defining the overall relationships between those ingredients. Two dynamic agents were designed to provide tailored advice and suggestions on both local and global aspects of the model. They have been denominated: *What can I do next*? (local) and *Cross builder* help (global).

At the local level, help is generated on the learner's actual model-building activity. The help facility analyses the input of the learner within the active builder and guides the learner by providing a set of possible subsequent actions. Also context-sensitive help is given which focuses on the specific request for guidance from the learner. For instance, if the learner selects a *quantity* in the model fragment builder and then selects *What can I do next?*, only guidance regarding that primitive will be given. Figure 2 shows an example of help (RHS) given in the Structure Builder context (LHS). In this example the agent gives three advice options (inferred by using a set of rules specifying relationships between model ingredients): 'Create a structural relation', 'Create an attribute', and 'Work on the current selection' (because selections are made in the builder). Notice the first two are the *only* possible actions a learner can perform in the builder, given what s/he has already created. The learner has selected the third option, and the agent gives an explanation of that (RHS, agent window).



Structure Builder: What can be done

Advise

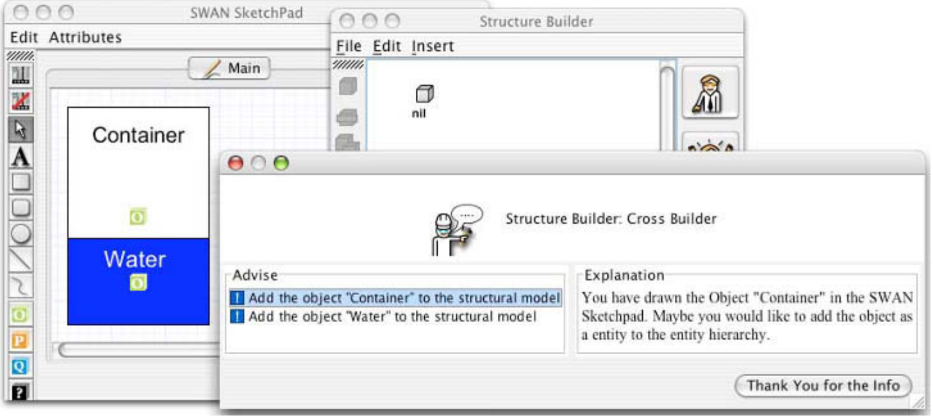
- Create a structural relation.
- Create an attribute.
- Work on the current selection

Explanation

You have selected two entities. You can add a structural relation between the selected entities or undo your selection.

Figure 2: ‘What can I do next?’ advice in the context of the Structure Builder.

Global feedback on the other hand is based on what the learner has previously constructed in *all* other builders then the one from which the help is requested. The idea is that ingredients are related and must somehow be re-used across different builders. If already defined model ingredients are not yet re-used adequately, and the re-use might be relevant to the builder from which the help is asked, then the agent will produce an advice on that. Sometimes many advices are possible. We have defined progress levels in order to generate contextual advices associated with each model-building step. Thus, the information gathered enables the help engine to create an ordered list of possible user actions applying to the specific model-building step. Figure 3 shows an example of a global feedback.



Structure Builder: Cross Builder

Advise

- Add the object "Container" to the structural model
- Add the object "Water" to the structural model

Explanation

You have drawn the Object "Container" in the SWAN Sketchpad. Maybe you would like to add the object as a entity to the entity hierarchy.

Figure 3: The *Cross Builder* agent refers to an object in the SWAN SketchPad

3. Design of the Experiment

A study was performed with three novices and four experts to assess the usefulness of the multi agent help system and the usability of the MOBUM user interface. The purpose of the novice/expert distinction was not to compare the performance of the two, but rather to ensure that an adequate range of users was covered. For this purpose, the participants were given tasks that corresponded to their capabilities. The task for the novices was to determine the effect of ‘food intake’ and ‘physical exercise’ on the ‘weight of Garfield’. Experts were asked to construct a simulation model of the two-tank system (U-tube). The participants received documentation concerning the assignment, a short explanation of the employed qualitative modelling terms, and a brief introduction to the MOBUM environment. Each session lasted one hour. In both situations (novices and experts) a drawing, illustrating the situation the participants should model, was available in the SWAN SketchPad, the drawing tool of MOBUM.

All computer actions as well as verbal data for each of the sessions were recorded on video. Two types of data were used to evaluate MOBUM: screen information and the verbal utterances of the participants. Participants were asked to think-aloud as much as possible, providing us with valuable information regarding the reasoning underlying the actions taken during the model-building task.

In order to measure the usefulness of the help system, we observed at which moments an agent was requested and if the given feedback was sufficient for clearing the doubts of the subjects about the problem at hand. Additionally, the questions posed by the participants to the experiment leader were analysed to verify whether they were in principle covered by the implemented help system (in which case they could just as well have been solved by the help agents!). While the participant completed each task, the experiment leader noted the number of times an agent was used. In order to measure the participants' performance, the models they created were compared to existing models created by experts.

A second study was performed especially to compare the two model-building environments, MOBUM and HOMER (Table 2). The goal was to evaluate whether the new prototype was more effective and if it would be more appreciated by the users. 28 first-year Psychology students participated in this study. None of the participants had knowledge about building qualitative models as well as about the two systems. The participants were randomly divided into two groups of 14. One of the groups started working with MOBUM for one hour and then changed to HOMER using it for 30 minutes. For the other group the order of the two programs was reversed. The assignment consisted also of building a *Garfield model* using each one of the two systems. The participants were then asked to fill out questionnaires on MOBUM (QM), HOMER (QH), and a third one on a direct comparison between the two systems (Com).

Table 2: Sequence of the questionnaires and tools in the comparison experiment.

Condition	**Tasks**			**Questionnaires**			
	8 min	60 min	30 min	15 min			
Mobum-Homer	Introduction	Mobum	Homer	QM	QH	Com	E
Homer-Mobum	Introduction	Homer	Mobum	QH	QM	Com	E

4. Results and Evaluation

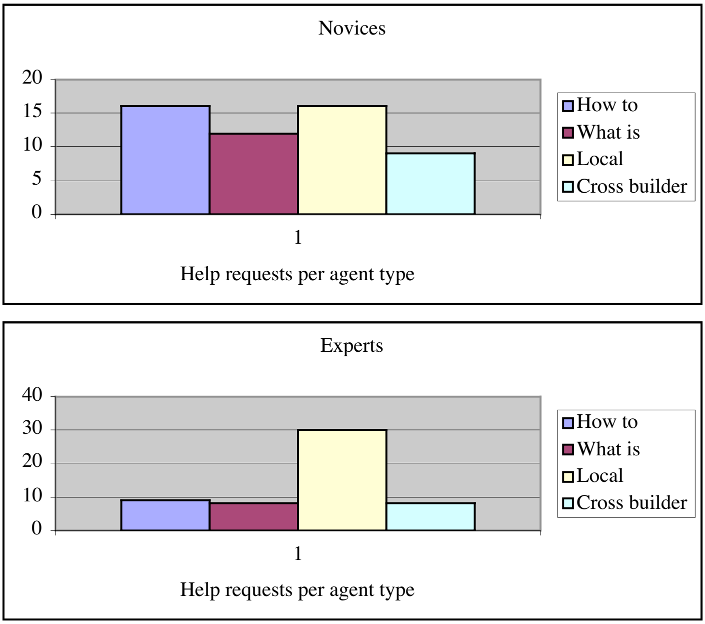
Table 3 summarises the usage of the agents by the participants. The novices requested help in all the builders and the requested help was of different kinds. Experts on the other hand needed help mainly in the context of model fragments and they accessed the local agent most frequently. This may be explained by the fact that creating a model fragment involves manipulation of all the single model ingredients created previously, as well as determining relations between them.

Without exception all novices found the agents useful and essential. The help facility was essential in aiding the participants in solving conceptual problems. For example, a participant wrongly specified *quantities* as *entities* using the Structure Builder. When specifying a model fragment, the participant realised that it was impossible to define *dependencies* between *entities* (they can only be defined between *quantities)*. So, the participant backtracked and consulted the agent to understand what had been done wrongly. In doing so, the participant learned what the mistake was.

Another participant had no knowledge about (qualitative) modelling and consequently also no understanding of *points* and *intervals* in a quantity space. But during the process of creating a quantity space, the participant learned about them. It took the participant 15 minutes to specify the first quantity space, 2 minutes for the second, and 30 seconds for the third. In yet another case, after consulting the agents, the participant found the explanation about derivatives and understood their meaning. Later, the participant returned and used the concepts correctly.

The experts did not seem to use the agents to solve problems. When the experts got stuck, they consulted the experiment leader. However, the participants might as well have consulted the agents, as their problems could have been dealt with using the agent-based help facility. Experts seem to use the agents to assess the help potential by trying the help in different situations. However, when trying the agents, the advices inspired them. Another support feature frequently consulted was the SWAN SketchPad, the drawing tool of MOBUM, which contained the U-tube drawing. The participants were consulting the drawing in order to verify if their model included all the details presented in the drawing.

Table 3: Usage of agents by novices and experts



Experts had only a few problems that specifically related to the MOBUM user interface. In our study with HOMER 67 problems were observed while in MOBUM only 10 problems were observed. These results indicate that the features implemented in MOBUM are insightful and effectively support modellers in building their models.

4.1 Results of the Comparison Study

In both situations, HOMER-MOBUM and MOBUM-HOMER there was a strong (and significant) preference for MOBUM over HOMER. For instance, the results of the comparative questionnaire clearly show a significant preference for MOBUM over HOMER ($z=4.4$, $p<0.0005$). Even when only the first tool ($z=2.7$, $p=0.007$) or the second tool ($z=3.6$, $p < 0.0005$) is measured there was a significant preference for MOBUM over HOMER. A variance analysis was performed to find out if the order had an influence on the results of the experiment. This was not the case.

The results for the measure of productivity of both tools did not prove that MOBUM was more effective. We expected that by being more easy to use and giving more guidance, a difference in productivity would emerge. But, there was a high variance among the participants and therefore no strong conclusions can be drawn with respect to this issue. For additional details see [1].

5. Conclusions and Discussion

This paper discusses a multi-agent help system that supports learners in building qualitative knowledge models using diagrammatic representations. Being able to create such *conceptual* models (concept maps) may help learners in understanding how and why systems behave as they do. The multi-agent help system is implemented as a part of MOBUM, a workbench for building, simulating, and inspecting qualitative models. The agents are personified and provide context-sensitive help. They provide general support on for instance the model-building ontology, as well as tailored feedback addressing the individual needs of learners.

A study was performed to assess the usefulness of the multi-agent support module. The results are encouraging. Most of the problems the participants encountered were (or could have been) solved by consulting the agents, which reinforces the idea that MOBUM in fact supports the model-building process. A second study was performed to compare MOBUM and HOMER, an earlier developed model-building tool. Due to the large variation in the models created during the experiment we cannot prove that MOBUM is more effective. However, it is safe to conclude that the multi-agent help module effectively influenced the appreciation of the tool: subjects evaluated MOBUM significantly more positive.

Future work could focus on a number is issues. Some initial work has been done on using our model-building workbenches in classroom situations [11]. Significantly more effort is needed to actually fit this new approach to science teaching and learning in currently used curricula. Related is the fact that MOBUM is a prototype system. Although it has all the required functionality, it is not fully stable as software package. For use in classrooms this needs to be addressed.

References

[1] Bessa Machado, V. (2004) Supporting the Construction of Qualitative Knowledge Models. Ph.D. Thesis, University of Amsterdam, Amsterdam.

[2] Bessa Machado, V. and Bredeweg, B. (2003) Building Qualitative Models with HOMER: A Study in Usability and Support. In: P. Salles and B. Bredeweg (Eds.), Proceedings of the 17th International workshop on Qualitative Reasoning, pages 39-46, Brasilia, Brazil, August 20-22.

[3] Biswas, G., Schwartz, D., Bransford, J. and The Teachable Agents Group at Vanderbilt. (2001) Technology Support for Complex Problem Solving: From SAD Environments to AI. In: K. Forbus and P. Feltovich (Eds.). Smart Machines in Education. AAAI Press/MIT Press, Menlo Park California, USA.

[4] Bredeweg, B. (1992) Approaches to Qualitative Reasoning. Ph.D. thesis, University of Amsterdam, Amsterdam.

[5] Bredeweg, B. and Forbus, K. (2003) Qualitative Modeling in Education. AI Magazine, 24(4):35-46.

[6] Collins, A. (1996) Design issues for learning environments. In: S. Vosniadou, E.D. Corte, R. Glaser and H. Mandl (Eds.), International perspectives on the design of technology-supported learning environments, pages 347-362, Lawrence Erlbaum, Mahwah, New Jersey.

[7] Forbus, K.D., Carney, K., Harris, R. and Sherin, B.L. (2001) A qualitative modeling environment for middle-school students: A progress report. In: G. Biswas (Ed.), Proceedings of the 15th International Workshop on Qualitative Reasoning, pages 65-72, St. Mary's University, San Antonio, Texas.

[8] Novak, J.D. and Gowin, D.B. (1984) Learning how to learn. Cambridge University Press, New York, New York.

[9] Reif, F. and Larkin, J.H. (1991) Cognition in scientific and everyday domains: comparison and learning implications. Journal of Research in Science Teaching, 28:733-760.

[10] Shimoda, T.A., White, B.Y. and Frederiksen J.R. (2002) Student goal orientation in learning inquiry skills with modifiable software advisors. Science Education, 86:244-263.

[11] Werf, v.d. R. (2003) The use of qualitative modelling and simulation tools in high school education: an engineering study. Master thesis, University of Amsterdam, Amsterdam.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Analyzing Completeness and Correctness of Utterances Using an ATMS

Maxim Makatchev [1] and Kurt VanLehn
Learning Research and Development Center, University of Pittsburgh

Abstract. Analyzing coverage of a student's utterance or essay (completeness) and diagnosing errors (correctness) can be treated as a diagnosis problem and solved using a well-known technique for model-based diagnosis: an assumption-based truth maintenance system (ATMS). The function-free first-order predicate logic (FOPL) representation of the essay is matched with nodes of the ATMS that are then analyzed for being within the sound part of the closure or relying on a particular misconception. If the matched nodes are sound they are analyzed for representing a particular required physics statement. If they do not represent the required statement, a neighborhood (antecedent and consequent nodes within N inference steps) of these nodes can be analyzed for matching the statement, to give a measure of how far the student utterance is, in terms of a number of inferences, from the desired one.

Keywords. Dialogue-based intelligent tutoring systems, formal methods in natural language understanding, ATMS

1. Introduction

Analyzing student input to an intelligent tutoring system for coverage (completeness) and errors (correctness) is essential for generating adequate feedback. When the student input is spoken or typed natural language (NL), analysis of the input becomes a significant problem. While statistical methods of analysis in many cases are sufficient [2], our tutoring system, Why2-Atlas [11], must analyze coverage and errors at a fine grain-size so that it can pinpoint students' mistakes and help students learn from them. This finely detailed analysis requires a large number of classes whose representatives have nearly the same bags of words and syntactic structures. This makes it very difficult for statistical classifiers to determine which classes best fit the student's input. Thus, Why2-Atlas is relying increasingly on non-statistical NLU in order to produce an adequately detailed analysis of student input.

In previous work [6], we demonstrated the feasibility of using an abductive reasoning back-end for analyzing students' NL input. A major part of this work involved defining and refining the knowledge representation language. As the development progressed, it became clear that adequate tutoring depended on being able to make fine distinctions, so the language became increasingly fine-grained. As the granularity decreased, the number

[1] Correspondence to: Maxim Makatchev, LRDC, 3939 O'Hara Street, Pittsburgh, PA 15260, USA. Tel.: +1 412 624 7498 ; Fax: +1 412 624 7904; E-mail: maxim@pitt.edu.

of inferences required to connect utterances increased. The abductive reasoning backend would make these inferences at run-time using the Tacitus-lite+ theorem prover. As the number of inferences to be made at run-time increased, it became more difficult to provide a guaranteed bound on the response time of the tutoring system.

In order to improve the response time of Why2-Atlas and to increase the maintainability of the knowledge base, we have switched to precomputing as much of the reasoning as possible. In particular, Why2-Atlas now precomputes all the reasoning that depends only on the problem and not on the student's solution to the problem. Reasoning that depends on the student input is of course still done at runtime. Because so much reasoning is done in advance, we can check each problem's precomputed reasoning thoroughly in order to guarantee that no flaws have crept into the knowledge base.

For this purpose, we adopted an augmented assumption-based truth maintenance system (ATMS) to precompute the desired reasoning [1]. Essentially, the precomputation requires computing the deductive closures of a set of rules of physics (e.g., "zero net force implies zero acceleration") and a set of propositions representing a particular problem (e.g., "the truck has a larger mass than the car"). However, our knowledge representation includes rules for student misconceptions, such as "zero force implies velocity decreases." Including both buggy rules and correct ones in the same deductive closure introduces inconsistencies. Thus, each student misconception is treated as an assumption (in the ATMS sense), and all conclusions that follow from it are tagged with a label that includes it as well as any other assumptions/misconceptions needed to derive that conclusion. This labeling essentially allows the ATMS to represent many interwoven deductive closures, each depending on different misconceptions, without inconsistency.

This also makes is much easier to check the precomputed reasoning for flaws. By examining the labels, one can easily figure out how a conclusion was reached, which facilitates debugging the knowledge base. Moreover, it allows us to automate regression testing. Whenever a significant change is made to the knowledge base, one compares the newly computed conclusions to those saved just before making a change. Similar advantages have driven other ITS projects to use precomputed reasoning as well [12,9].

This paper begins by reviewing the NLU task of Why2-Atlas and its knowledge representation in Sections 2 and 3. We then discuss the design choices for the ATMS (Section 4) and the structure of the completeness and correctness analyzer (Section 5). We end with the preliminary evaluation results in Section 6 and the conclusions in Section 7.

2. Role of NLU in Why2-Atlas tutoring system

The Why2-Atlas tutoring system is designed to encourage students to write their answers to qualitative mechanics problems along with detailed explanations supporting their arguments [11]. A typical problem and a student explanation is shown in Figure 1.

Each problem has an ideal "proof" designed by expert physics tutors that contains steps of reasoning, i.e. facts and their justifications, and ends with the correct answer. The proof for the Clay Balls problem from Figure 1 is given in Figure 2. Not all of the proof facts and justifications are required to be present in an acceptable student essay. The task of the NLU module is to identify whether the required points have been mentioned and whether any of the essay propositions are related to a set of known common misconceptions.

Problem: A heavy clay ball and a light clay ball are released in a vacuum from the same height at the same time. Which reaches the ground first? Explain.
Explanation: Both balls will hit at the same time. The only force acting on them is gravity because nothing touches them. The net force, then, is equal to the gravitational force. They have the same acceleration, g, because gravitational force=mass*g and f=ma, despite having different masses and net forces. If they have the same acceleration and same initial velocity of 0, they have the same final velocity because acceleration=(final-initial velocity) elapsed time. If they have the same acceleration, final, and initial velocities, they have the same average velocity. They have the same displacement because average velocity=displacementtime. The balls will travel together until the reach the ground.

Figure 1. The statement of the problem and a verbatim student explanation.

Step	Proposition	Justification
1	Both balls are near earth	Unless the problem says otherwise, assume objects are near earth
2	Both balls have a gravitational force on them due to the earth	If an object is near earth, it has a gravitational force on it due to the earth
3	There is no force due to air friction on the balls	When an object is in a vacuum, no air touches it
4	The only force on the balls is the force of gravity	Forces are either contact forces or the gravitational force
5	The net force on each ball equals the force of gravity on it	[net force = sum of forces], so if each object has only one force on it, then the object's net force equals the force on it
6	**Gravitational force is w = m*g for each ball**	**The force of gravity on an object has a magnitude of its mass times g, where g is the gravitational acceleration**
⋮	⋮	⋮
18	**The balls have the same initial vertical position**	given
19	The balls have the same vertical position at all times	[Displacement = difference in position], so if the initial positions of two objects are the same and their displacements are the same, then so is their final position
20	**The balls reach the ground at the same time**	

Figure 2. A fragment of an ideal “proof” for the Clay Balls problem from Figure 1. The required points are in bold.

After the essay analysis is complete the tutoring feedback may be a dialogue that addresses missing required points or erroneous propositions. During a dialogue an analysis similar to that performed during the essay stage may be required for some student turns: does the student's dialogue turn include a required point or is it related to a known misconception.

3. Knowledge representation

The difficulty of converting unconstrained natural language into a formal representation is one of the main obstacles to using formal reasoning techniques for NLU. We designed FOPL representation that is expressive enough to cover the physics domain propositions we are interested in, and is able to preserve formal and informal descriptions of the domain concepts (for example, “the force is downward” versus “the horizontal component of the force is zero and the vertical component is negative”, “the balls’ positions are the same” versus “the balls move together”) [5], and can incorporate algebraic expressions (for example, “F=ma”). This relatively fine granularity of representation for degrees of formality in NL is useful for providing more precise tutoring feedback, and can be generated by language understanding approaches that include statistical classifiers [3].

To demonstrate the flexibility of the KR with an example, we include a few slightly abridged representations below:

“the balls’ positions are the same”

```
(position p1 big-ball ?comp1 ?d-mag1 ?d-mag-num1
      ?mag-zero1 ?mag-num1 ?dir1 ?dir-num1 ?d-dir1 ?time1 ?time2)
(position p2 small-ball ?comp1 ?d-mag1 ?d-mag-num1
      ?mag-zero1 ?mag-num1 ?dir1 ?dir-num1 ?d-dir1 ?time1 ?time2)
```

“the balls move together”

```
(motion m1 big-ball ?comp2 ?traj-shape2 ?traj-speed2 ?d-mag2
    ?d-mag-num2 ?mag-zero2 ?dir2 ?dir-num2 ?d-dir2 ?time3 ?time4)
(motion m1 small-ball ?comp ?traj-shape ?traj-speed ?d-mag2
    ?d-mag-num2 ?mag-zero2 ?dir2 ?dir-num2 ?d-dir2 ?time3 ?time4)
```

In these examples the equality of arguments of two predicates is represented via the use of shared variables.

4. ATMS design

ATMS’s have been used for tasks that are closer to the front end of the NLU processing pipeline such as for parsers that perform reference resolution (e.g. [7]), but there are few systems that utilize an ATMS at deeper levels of NLU [4,13]. In our view, given that a formal representation of student input is obtained, the task of analyzing its completeness and correctness can be treated as a diagnosis problem and solved by methods of model-based diagnosis. In this section we describe in detail the ATMS we designed for the task of diagnosing formal representations of NL utterances.

For the description of ATMS features below we adopt the terminology from [1]:

- *Premises* are givens of the physics problem (“initial positions of balls are the same,” etc.)
- *Assumptions* are statements about student beliefs in a particular misconception (“Student believes that heavier objects fall faster”).
- *Deduction rules* are the rules of inferences in the domain of mechanics (“zero force implies zero acceleration”).

- *Nodes* are the atoms of the FOPL representation that are derived from the givens and assumptions via forward chaining with the deduction rules.
- *Labels* are assumptions that were made on the way to derive the particular node.
- *Environment* is a consistent set of assumptions that are sufficient to infer a node.

Our implementation of the ATMS relaxes the usual requirement of consistency of the deductive closure, because in our context students may hold inconsistent beliefs. While this certainly increases the size of the deductive closure, it may potentially provide a better explanation of the student's actual reasoning. The degree of ATMS consistency needed to best match with the observed student's reasoning is a topic we will explore during a future evaluation.

5. Completeness and correctness analyzer Cocoro

All domain statements that are potentially required to be recognized in the student's explanation or utterances are divided into principles and facts. The principles are versions of general physics (and "buggy physics") principles that are either of a vector form (for example, "F=ma") or of a qualitative form (for example,"if total force is zero then acceleration is zero"), while facts correspond to concrete instantiations of the principles (for example, "since there is no horizontal force on the ball its horizontal acceleration is zero") or to derived conclusions (for example, "the horizontal acceleration of the ball is zero"). As a natural consequence of the fact that the ATMS deductive inferences are derived from the problem givens, which are instantiated facts, the ATMS includes only facts. Therefore the recognition of both general principles and facts must be restricted to the actual input representations, while the ATMS is used only for recognizing and evaluating the correctness of facts closely related to the student's utterances, as shown in Figure 3 and elaborated below.

The nodes of the ATMS that match the representation of the input utterance are analyzed for correctness by checking whether their labels contain only environments with buggy assumptions. If there are no environments that are free of buggy assumptions in the label of the node, the node can only be derived using one of the buggy assumptions and therefore represents a buggy fact. These buggy assumptions are then reported to the tutoring-system strategist for possible remediation. If the nodes are correct (labels contain assumption-free environments) they are matched with required statements and the list of matched statements is then reported to the tutoring-system strategist for possible elicitation of any missing points. Additionally, a neighborhood of radius N (in terms of a graph distance) of the matched nodes can be analyzed for whether it contains any of the required principles to get an estimate of the proximity of a student's utterance to a required point.

For example, given the formal representation for the student utterance "the balls have the same vertical displacement," Cocoro attempts to both directly match it with stored statement representation (the right branch in the diagram in Figure 3) and find a set of matching nodes in the ATMS (the left branch in the diagram in Figure 3). If the direct match succeeds this already provides information about whether the student statement is correct or not. If the direct match fails, namely we do not have a stored representation for this fact, then we arrive at a conclusion about the correctness of the student's statement by examining the labels of the ATMS nodes that matched the input statement, if there are

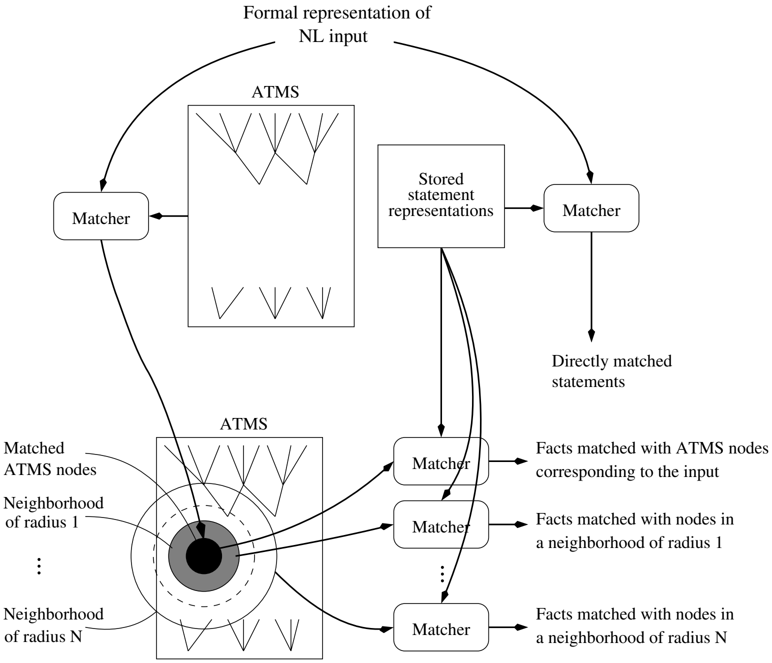

Formal representation of NL input
ATMS
Matcher
Stored statement representations
Matcher
Directly matched statements
ATMS
Matched ATMS nodes
Neighborhood of radius 1
⋮
Neighborhood of radius N
Matcher
Matcher
⋮
Matcher
Facts matched with ATMS nodes corresponding to the input
Facts matched with nodes in a neighborhood of radius 1
Facts matched with nodes in a neighborhood of radius N

Figure 3. Completeness and correctness analyzer Cocoro. A description of the diagram is in the text.

any (represented by the black circle in the ATMS block in Figure 3). The neighborhoods of the matched ATMS nodes can also be examined for matching with stored statements. For example, the nodes for the stored required fact “The balls have the same vertical position” would be within distance 1 from the set of nodes that matched the student utterance “The balls have the same vertical displacement.” This information can lead to an encouraging feedback to let the student know that she is one inference away from the desired answer.

Formal representations are matched by a version of a largest common subgraph-based graph-matching algorithm (due to the need to account for cross-referencing atoms via shared variables) proposed in [10], that is particularly fast when one of the graphs to match is small and known in advance, as is the case with all but one of the Matcher blocks shown in Figure 3. In case of the Matcher for the formal representation of the NL input, which is not known in advance, the set of ATMS nodes is known but large. For this case we settle for an approximated evaluation of the match via a suboptimal largest common subgraph.

6. Preliminary evaluation

The Cocoro analyzer is being deployed in an ongoing evaluation of the full Why2-Atlas tutoring system. Figure 4 shows results of classifying 135 student utterances for two physics problems using only direct matching (66 utterances with respect to 46 stored representations and 69 utterances with respect to 44 stored representations). To generate

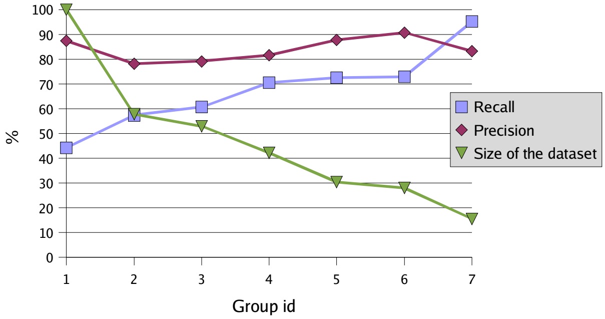


Figure 4. Average recall and precision of utterance classification by Cocoro. The size of a group of entries is shown relative to the size of the overall data set. Average processing time is 0.46 seconds per entry on a 1.8 GHz Pentium 4 machine with 2Gb of RAM.

these results, the data is divided into 7 groups based on the quality of conversion of NL to FOPL, such that group 7 consists only of perfectly formalized entries, and for $1 \leq n \leq 6$ group n includes entries of group $n+1$ and additionally entries of somewhat lesser representation quality, so that group 1 includes all the entries of the data set. The flexibility of the matching algorithm allows classification even of utterances that have mediocre representations, resulting in 70.6% average recall and 81.6% average precision for 42.2% of all entries (group 4). However, large numbers of inadequately represented utterances (at least 47%) result in 44.3% average recall and 87.4% average precision for the whole data set (group 1). Note that Cocoro analyzes only utterances for which *some* representation in FOPL has been generated. Figure 4 does not include data on utterances for which no formal representation has been generated; such utterances are classified relying on a statistical classifier only [8].

At the same time we are investigating the computational feasibility of utilizing the full Cocoro analyzer with ATMS. One of the concerns is that as the depth of the inferencing increases, ATMS size can make real-time matching infeasible. Our results show that an ATMS of depth 3, generated using just 11 physics inference rules, and containing 128 nodes, covers 55% of the relevant problem facts. It takes about 8 seconds to analyze an input representation consisting of 6 atoms using an ATMS of this size, which is a considerable improvement over the time required for the on-the-fly analysis performed by the Tacitus-lite+ abductive reasoner [6]. The knowledge engineering effort needed to increase the coverage is currently under way and involves enriching the rule base.

7. Conclusions

In this paper we described how we alleviate some of the performance and knowledge engineering drawbacks associated with using an on-the-fly abductive reasoner by deploying a precomputed ATMS as a back-end for an analyzer of completeness and correctness of student utterances. Besides the improvement in time response, the ATMS-based analysis provides the additional possibility of evaluating an “inferential neighborhood” of the student’s utterance which we expect to be useful for providing more precise tutoring feedback. The preliminary evaluation provided encouraging results suggesting that we can successfully deploy the ATMS-based reasoner as an NLU back-end of the Why2-Atlas tutoring system.

Acknowledgements

This research has been supported under NSF grant 0325054 and ONR grant N00014-00-1-0600. The authors would like to thank the Natural Language Tutoring group for their work on the Why2-Atlas system, in particular Pamela Jordan, Brian 'Moses' Hall, Umarani Pappuswamy, and Michael Ringenberg.

References

[1] Kenneth D. Forbus and Johan de Kleer, editors. *Building Problem Solvers*. MIT Press, Cambridge, Massachusetts; London, England, 1993.

[2] Arthur C. Graesser, Peter Wiemer-Hastings, Katja Wiemer-Hastings, Derek Harter, Natalie Person, and the TRG. Using latent semantic analysis to evaluate the contributions of students in autotutor. *Interactive Learning Environments*, 8:129–148, 2000.

[3] Pamela W. Jordan, Maxim Makatchev, and Kurt VanLehn. Combining competing language understanding approaches in an intelligent tutoring system. In *Proceedings of Intelligent Tutoring Systems Conference*, volume 3220 of *LNCS*, pages 346–357, Maceió, Alagoas, Brazil, 2004. Springer.

[4] Yasuyuki Kono, Takehide Yano, Tetsuro Chino, Kaoru Suzuki, and Hiroshi Kanazawa. Animated interface agent applying ATMS-based multimodal input interpretation. *Applied Artificial Intelligence*, 13(4-5):487–518, 1999.

[5] Maxim Makatchev, Pamela W. Jordan, Umarani Pappuswamy, and Kurt VanLehn. Abductive proofs as models of students' reasoning about qualitative physics. In *Proceedings of the 18th International Workshop on Qualitative Reasoning*, pages 11–18, Evanston, Illinois, USA, 2004.

[6] Maxim Makatchev, Pamela W. Jordan, and Kurt VanLehn. Abductive theorem proving for analyzing student explanations to guide feedback in intelligent tutoring systems. *Journal of Automated Reasoning, Special issue on Automated Reasoning and Theorem Proving in Education*, 32:187–226, 2004.

[7] Toyoaki Nishida, Xuemin Liu, Shuji Doshita, and Atsushi Yamada. Maintaining consistency and plausibility in integrated natural language understanding. In *Proceedings of COLING-88*, volume 2, pages 482–487, Budapest, Hungary, 1988.

[8] Umarani Pappuswamy, Dumisizwe Bhembe, Pamela W. Jordan, and Kurt VanLehn. A multi-tier NL-knowledge clustering for classifying students' essays. In *Proceedings of 18th International FLAIRS Conference*, 2005.

[9] S. Ritter, S. Blessing, and L. Wheeler. User modeling and problem-space representation in the tutor runtime engine. In *Proceedings of the 9th International Conference on User Modelling*, volume 2702 of *LNAI*, pages 333–336. Springer, 2003.

[10] Kim Shearer, Horst Bunke, and Svetha Venkatesh. Video indexing and similarity retrieval by largest common subgraph detection using decision trees. *Pattern Recognition*, 34(5):1075–1091, 2001.

[11] Kurt VanLehn, Pamela Jordan, Carolyn Rosé, Dumisizwe Bhembe, Michael Böttner, Andy Gaydos, Maxim Makatchev, Umarani Pappuswamy, Michael Ringenberg, Antonio Roque, Stephanie Siler, and Ramesh Srivastava. The architecture of Why2-Atlas: A coach for qualitative physics essay writing. In *Proceedings of Intelligent Tutoring Systems Conference*, volume 2363 of *LNCS*, pages 158–167. Springer, 2002.

[12] Kurt VanLehn, Collin Lynch, K. Schultz, Joel Shapiro, R. H. Shelby, Linwood Taylor, D. J. Treacy, Anders Weinstein, and M. C. Wintersgill. The Andes physics tutoring system: Lessons learned (under review). *Unpublished manuscript*.

[13] Uri Zernik and Allen Brown. Default reasoning in natural language processing. In *Proceedings of COLING-88*, volume 2, pages 801–805, Budapest, Hungary, 1988.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Modelling Learning in an Educational Game

Micheline Manske, Cristina Conati
Department of Computer Science, University of British Columbia,
Vancouver, BC, V6T1Z4, Canada
{manske, conati}@cs.ubc.ca

Abstract. We describe research on data-drive refinement and evaluation of a probabilistic model of student learning for an educational game on number factorization. The model is to be used by an intelligent pedagogical agent to improve student learning during game play. An initial version of the model was designed based on teachers' advice and subjective parameter settings. Here we illustrate data-driven improvements to the model, and we report results on its accuracy.

1. Introduction

A student model is one of the fundamental components of an intelligent learning environment [11], and much research has been devoted to creating student models for various types of computer based support. However, little work exists on student modelling for a relatively new type of pedagogical interaction, educational computer games (edu-games from now on). In this paper, we describe the design and evaluation of a student model to assess student learning during the interaction with Prime Climb, an edu-game for number factorization.

The main contribution of this work is a step toward providing intelligent computer based support to learning with edu-games. Providing this support is both extremely valuable and extremely challenging. It is valuable because, although there is overwhelming evidence that even fairly simple edu-games can be highly motivating, there is little evidence that these games, no matter how sophisticated they are, can actually trigger learning, unless they are integrated with ad hoc supporting activities [5,9,6]. This is because many students manage to successfully play these games without necessarily having to reason about the underlying domain knowledge. We argue that individualized support based on careful assessment of student learning during game playing can help overcome this limitation and make edu-games an effective new form of learning.

Providing this support is challenging because it requires careful tradeoffs between fostering learning and maintaining positive affective engagement. Thus, it is crucial to have accurate models of both student learning and affect. Creating these models is hard, however, because it necessitates understanding about cognitive and affective processes on which there is very little knowledge, given the relative novelty of games as educational tools. In [2] we present a model of student affect for the Prime Climb edu-game. Here we focus on the model of student learning. In particular, we describe the data-drive refinement and evaluation of an initial model based on expert knowledge and subjective judgements, previously described in [3].

There is increasing research in learning student models from data (e.g., [1,4,7]), but most of this research has focused on student models for more traditional ITS systems. An exception is [8], which describes a student model learned from data for a game designed to address common misconceptions about decimal numbers. The data used in [8] come from students' performance on a traditional test to detect decimal number misconceptions. Thus, the model parameters learned from these data, (e.g., the probability of an error of distraction (*slip*) or a lucky guess), do not reflect the actual relationship between student performance and knowledge during game playing. This relationship is likely to be different than in traditional tests. Several studies have shown that students can be successful game players by learning superficial heuristics rather then by reasoning about the underlying domain knowledge. Furthermore, students may make more slips during game playing, because they are distracted by the game aspect of the interaction. In the work presented here, the data used to learn the student model comes from interaction with Prime Climb. Thus, the model parameters provide us with insights on how students learn and interact with this type of educational system, in itself a contribution given the relative lack of understanding of these mechanisms.

In the rest of the paper, we first introduce the Prime Climb game and an initial version of its student model (both described in more details in [3]). Next, we present a study to evaluate this model's accuracy. We then describe a data-drive refinement of the model, assess its accuracy and analyze the sensitivity to its various parameters. Finally, we introduce a further improvement with the modelling of common factoring, and compare the three student models.

2. The Prime Climb Game and Initial Student Model

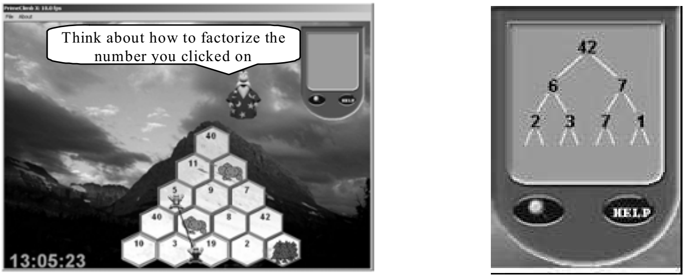


Figure 1a: The Prime Climb Interface **b:** A factor tree displayed in the PDA

In Prime Climb (devised by the EGEMS group at the University of British Columbia) students in 6th and 7th grade practice number factorization by pairing up to climb a series of mountains. Each mountain is divided into numbered sectors (see Figure 1a), and players must try to move to numbers that do not share common factors with their partner's number, otherwise they fall. To help students, Prime Climb includes the Magnifying Glass, a tool that allows players to view the factor tree for any number on a mountain. This factor tree is shown in the PDA displayed at the top right corner of the game (see Figure 1b).

Each student also has a pedagogical agent (Figure 1a) which provides individualized support, both on demand and unsolicited, when the student does not seem to be learning from the game (see [3] for more details on the agent's behaviours). To provide appropriate interventions, the agent must have an accurate model of student learning. However, this modelling task involves a high level of uncertainty because, as we discussed earlier, game performance tends to be a fairly unreliable reflection of student knowledge. We use Dynamic Bayesian networks (DBNs) to handle this uncertainty.

A DBN consists of *time slices* representing relevant temporal states in the process to be modelled. In Prime Climb, there is a DBN for each mountain that a student climbs (the *short-term student model*). A time slice is created in this network after every student action, to capture the evolution of student knowledge as the climb proceeds. Each short term model includes the following random binary variables:

- *Factorization (F) Nodes:* each factorization node F_x represents whether the student has mastered the factorization of number x down to its prime factors.
- *Knowledge of Factor Tree (KFT) Node:* models knowledge of the factor tree representation.

- *Click Nodes*: each click node C_x models the correctness of a student's click on number *x*.
- *Magnification (Mag) Nodes :* each Mag_x node denotes using the magnifying glass on number *x*.

The network for a given mountain includes F nodes for all its numbers, F nodes for their factors, and the KFT node. Click and Mag nodes are introduced in the model when the corresponding actions occur, and are immediately set to one of their values.

Figure 2 illustrates the structure that we used in the first version of the model to represent the relations between factorization and click nodes[1]. A key assumption underlying this structure, derived from mathematics teachers, is that knowing the prime factorization of a number influences the probability of knowing the factorization of its factors, while the opposite is not true. It is hard to predict if a student knows a number's factorization given that s/he knows how to factorize its non-prime factors.

To represent this assumption, F nodes are linked as parents of nodes representing their non-prime factors. The conditional probability table (CPT) for each non-root F node (e.g. F_X in Figure 2a) is defined so that the probability of the node being known is high when all the parent F nodes are true, and decreases proportionally with the number of unknown parents. The action of clicking on number *x* when the partner is on number *k* is represented by adding a click node C_x as parent of nodes F_x and F_k (see Figure 2b). Thus, evidence coming from click actions is represented in the diagnostic rather than causal direction. This structure prevents evidence on a number *x* from propagating upwards to the numbers that contain it as a factor (e.g. F_Z in Figure 2b), thus respecting the insights provided by our teachers.

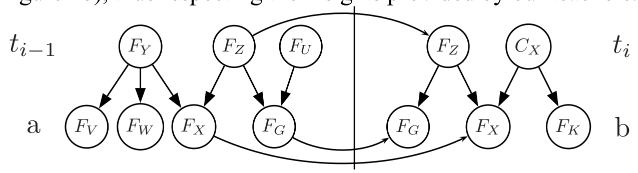


Figure 2. a: Factorization nodes, where Z=X*G and Y=V*W*X; **b:** Click action

Note that this model has two major limitations. The first is that it does not apportion blame for an incorrect click in a principled way. The two F nodes involved in a click should be conditionally dependent given the action, so that the node with the lower probability can be "blamed" more for the incorrect move. This dependency could be modelled by adding a link between the two F nodes (e.g, F_x and F_k in Fig. 2b), however this would increase the model's complexity so we chose not to. The second limitation is that the model does not include a node to explicilty represent knowledge of the common factor concept, which is a key component in playing the game successfully.

Although we were aware of these limitations, we wanted to investigate how far this relatively simple model would take us. In an initial study, the game with the agent giving help based on the above model generated significantly better learning than the game without agent [3]. However, the study was not designed to ascertain the role of the model in this learning. Hence, we ran a second study specifically designed to determine the model's accuracy.

2.1 Study for Model Evaluation

The study included data from 52 students in 6th and 7th grade. Each student played Prime Climb for approximately 10 minutes, with an experimenter as partner. All game actions were logged. Students were given identical pre and post-tests to gauge their factorization knowledge of 10 numbers frequently involved in the first two game levels, as well as their understanding

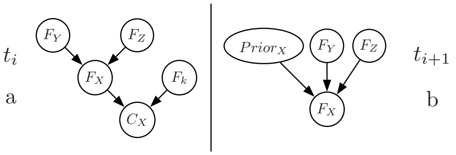
[1] We don't discuss the mechanisms to model learning through usage of the magnifying glass, because they are not involved in the model refinement process discussed here. See [3] for more details.

of the common factoring concept. We used the post-test answers to evaluate the model's assessment after game play (as explained in section 3.1). Despite an effort to fine-tune the model using data from the study, its accuracy was no better than chance (50.8%). This is not surprising, given the model limitations described above. The fact that agent condition showed significantly better learning indicates that even hints based on an almost random model are better than no hints at all. However, the fact that there was still large room for improvement in the post-tests of the agent-condition suggests that a more accurate student model may yield even more substantial learning gains. Thus, we set to improve our model to incrementally address the two limitations discussed earlier. This process resulted in two new versions of the model, both with parameters learned from data, which we illustrate in the following sections.

3. New Model – Causal Structure

One of the limitations of the original model is that it did not correctly apportion blame for incorrect moves. The new model uses a causal structure over click nodes to fix this problem. Each click node is added as child of the two F nodes involved in the click (see Figure 3a in contrast to Figure 2b). Thus, these nodes become conditionally dependent given a click and share the blame for an incorrect action proportionally to their probability.

F_x	F_k	P(Click=C)
K	K	*1-α*
K	U	*e_guess*
U	K	*e_guess*
U	U	*guess*



Prior	F_z	F_y	P(F_x=K)
K	K or U		1
U	K	K	*max*
	K	U	*max*/2
	U	K	*max*/2
	U	U	0

Figure 3. a: Click configuration at time t_i.; **b:** Roll-up on node F_x at time t_{i+1} when node F_x has two parents. K: known, U: unknown, C: correct

The three parameters needed to specify this configuration are α, *e_guess,* and *guess* (Figure 3a). The α parameter represents the probability of making an incorrect move despite knowing the factors of the relevant numbers, because of either a slip or lack of understanding of the common factoring concept. The *guess* parameter represents the probability of a correct move when both the numbers involved are unknown. The *e_guess* (educated guess) parameter is introduced to represents the possibility that it is easier to guess correctly when knowing the factorization of one of the numbers.

To reduce the computational complexity of evaluating the short-term model, at any given time we maintain at most two time slices in the DBN. This requires a process known as *roll-up*, i.e. saving the posterior probabilities of the slice that is removed (e.g., slice in Figure 3a) into the new slice that is created (e.g., slice in Figure 3b). Posterior probabilities of root nodes in the removed slice are simply saved as priors of the corresponding nodes in the new slice. For non-root nodes the process is more complicated, and requires different approaches for various network configurations [3,10]. The approach proposed here is as follows: for every non-root F node that needs to be rolled up (e.g. F_x in Figure 3a) we introduce an additional *Prior* node in the new time slice (e.g. $Prior_x$ in Figure 3b), and give it as a prior the posterior of the F node in the previous time slice.

The CPT for the F node in the new slice (see table for F_x in Figure 3b) is set up such that knowing the factorization in the previous time slice implies knowing the factorization in the current slice (i.e. we do not model forgetting). Otherwise, the probability of the node being known is 0 when all the parent F nodes are unknown, and increases proportionally with the number of known parents to a maximum of *max*, the probability that the student can infer the factorization of *x* by knowing the factorization of its parent nodes.

We now describe how we learn the parameters *α, e_guess, guess,* and *max* from data from the user study described in the previous section.

3.1 Setting Parameters from Data

When all the nodes involved in a given CPT are observable, the CPT values can be learned from frequency data. F nodes are not usually observable, however, we have pre and post-test assessment on 10 of these nodes for each of our 52 students. If we consider data points in which pre and post-test had the same answer, we can assume that the value of the corresponding F nodes remained constant throughout the interaction (i.e. no learning happened), and can use these points to compute the frequencies for the CPT entries involving *α, guess,* and *e_guess*. We found 58 such data points in our log files, yielding the frequencies in Table 1.

Table 1: Parameter estimates from click frequencies

Parameter	Freq	Points
a	0.23	44
e_guess	0.75	12
guess	0	2

As Table 1 shows, the frequency for the *α* parameter is based on 44 points, thus we feel confident fixing its value at 0.23. However, because we have far fewer points for the *e_guess* and *guess* parameters we must estimate these parameters in another manner. Similarly, we cannot use frequencies to set the *max* parameter as we do not have data on *Prior* nodes, which represent the (possibly changing) student knowledge at any given point in the interaction.

To select ideal values for *e_guess, guess* and *max* we attempt to fit the data to the answers students gave on post-tests. We fix the parameters to a specific triplet, feed each student's log file to the model, and then compare the model's posterior probabilities over the 10 relevant F nodes with the corresponding post-test answers. Repeating this for our 52 students yields 520 <*model prediction, student answer*> pairs for computing model accuracy. Since it would be infeasible to repeat this process for every combination of parameter values, we select initial parameter values by frequency estimates and intuition. Next we determine whether the model is sensitive to any of the three parameters, and if so, try other parameter settings. The values used initially for *e_guess* were {0.5,0.6,0.7}, chosen using Table 1 as starting point. For *guess* there are too few cases to base the initial values on frequencies, so we rely on the intuition that they should be less than or equal to the *e_guess* values, and thus use {0.4,0.5,0.6}. For *max* we use {0,0.2,0.4}. We try all 27 possible combinations of these values and chose the setting with the highest model accuracy.

To avoid over fitting the data, we perform 10-fold cross-validation by splitting our 520 data points to create 10 training/test folds. For each fold, we select the parameter triplet which yields the highest accuracy on the 90% of the data that forms that training set, and we report its accuracy on the 10% in the test set. We then select the parameter setting with the best training set performance across folds.

As our measure of accuracy, we chose (sensitivity + specificity)/2 [12]. Sensitivity is the percentage of known numbers that the model classifies as such; specificity is the percentage of unknown numbers classified as such. Thus, we need a threshold that allows us to classify model probabilities as *known* or *unknown*. To select an adequate threshold, we picked several different threshold values, and computed the average model accuracy on training set across all 10 folds and 27 parameter settings. The threshold yielding the highest average accuracy was 0.8 (see Table 2). Note that the standard deviation across folds is low, indicating that we are not over fitting the data.

Using a threshold of 0.8, the setting with best performance across all 10 folds (highest accuracy in all but one of the folds) was 0.5 for both *e_guess* and *guess* and 0 for *max*. The fact that the two guess parameters are high confirms previous findings that students can often perform well in educational games through lucky guesses or other heuristics not requiring correct domain knowledge.

Table 2: Average training set accuracy across folds by threshold

Threshold	Accuracy	Std. Dev.
0.4	0.624	0.010
0.5	0.697	0.009
0.65	0.753	0.007
0.8	0.772	0.007
0.95	0.725	0.006

The fact that they are equal indicates that there is no substantial difference in the likelihood of a lucky guess given different degrees of domain knowledge. The setting of 0 for *max* indicates that the teacher-suggested relation between knowing the factorization of a number and knowing the factorization of its non-prime factors may be too tenuous to make a difference in our model (more on this in the next section).

Using these settings, our model achieves an average test set accuracy of 0.776, with a sensitivity of 0.767, and a specificity of 0.786. This is a substantial improvement over the 0.508 accuracy of the old model.

3.2 Model Sensitivity to Individual Parameters

To investigate how sensitive our model is to each parameter, we fix two of the parameters and calculate the standard deviation of the model's accuracy across all three values of the third. This yields an average standard deviation of 0.002 for *e_guess*, 0.005 for *guess*, and 0.002 for *max*, indicating low sensitivity to small changes in these parameters. To rule out the possibility that the three values we initially chose for each parameter were not ideal, we try more extreme values (0.3 and 0.1 for *guess* and *e_guess*; 0.6 and 0.8 for *max*). All of them yielded worse accuracy, indicating that the model is sensitive to larger changes in these parameters. Slight variation of the α parameter also produced little change in accuracy, with more extreme values (0.1 and 0.5) decreasing accuracy. These results indicate that we were able to identify adequate value ranges for the parameters in our new model configuration, and that the model is not sensitive to small changes of these parameters in the given ranges. They also suggest that we could select a value slightly higher than 0 for the *max* parameter if we want to maintain the teacher-suggested relationship among F nodes in the model, or we can choose to ignore these relationships if we need to improve the efficiency of model update.

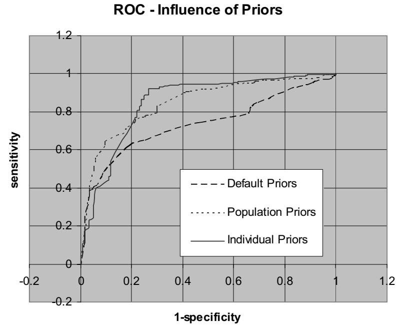


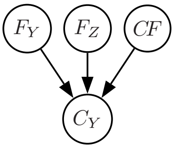
Figure 4: ROC curves comparing priors influence on sensitivity and specificity.

Finally, we analyzed the sensitivity of the model to the initial prior probability of F nodes. All results presented thus far have used *population* priors derived from frequencies over all students' pre-tests. We tried two more settings: (i) *Default*, which gives a prior of 0.5 for each F node; (ii) *Individual*, with priors derived from each student's pre-test answers. As the Receiver-Operator Curve (ROC) in Figure 4 show, population priors and individualized priors do better than default priors at most thresholds. However, the model can still have good performance even when accurate priors are not available (maximum accuracy is 0.717 for default, 0.776 for population, and 0.828 for individualized).

Although this new model has shown significant gains in accuracy, we wanted to see whether we could get further improvements by addressing the second limitation of the original model: omitting the concept of common factoring. We discuss its addition in the next section.

4. Modelling Common Factoring Knowledge

Because the model discussed above does not model common factor knowledge, when a student makes an incorrect move despite knowing the factorization of both numbers involved, the model can only infer that the student either made a slip or does not know the concept of common factors. This limits the system's capability to provide precise feedback based solely on model assessment. However, modelling common factor knowledge increases model complexity. To see how much we can be gained from this addition, we generated a new model that includes a common factor node (CF) as a parent of each click action (Figure 5). Note that the CPT entry corresponding to an incorrect action when all the parent nodes are known now isolates the probability of a slip. As before, the *guess* and *edu-guess* parameters in the CPT reflect potential differences in the likelihood of a lucky guess given different levels of existing knowledge.



CF	Fy	Fz	P(Click=C)
K	K	K	1-*slip*
K	K	U	*e_guess*
K	U	K	*e_guess*
K	U	U	*guess*
U	K or U		*guess*

Figure 5: Click configuration with common factor node

We use the same process described in the previous sections to set the parameters in the new model. Optimal threshold is again 0.8, while optimal parameter setting is 0.2 for *slip*, 0.6 for *e_guess* and *guess*, and 0 for *max*, showing good consistency with parameters in the model without CF node. Like that model, the new model is also not very sensitive to small changes in the parameters. Its average test set accuracy with population priors across all folds is 0.768 (SD 0.064) over F nodes and 0.654 for CF node (SD 0.08).

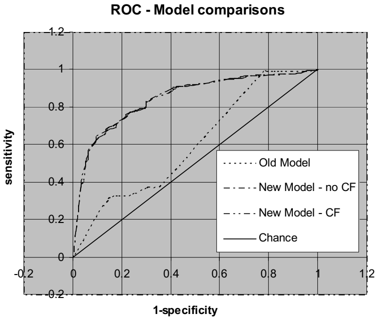


Figure 6: ROC curve comparisons of the three models and chance.

Figure 6 compares the accuracy of the three models and of a baseline chance model in assessing number factorization knowledge. As we can see, the accuracy of the assessment on number F nodes does not change considerably between the CF and no CF version. Furthermore, the assessment accuracy over CF is not very high. This may suggest that the addition of the CF node would not substantially increase the model's capability to support precise didactic interventions, and thus may not be worth the potential delays in model updates due to larger CPTs. However, two factors speak to the contrary. The first is that we have not seen these delays in our test runs. The second is that our current data may not be sufficient for accurate parameter learning in this more complex model, as it is suggested by the larger standard deviation of accuracy across folds compared to the no CF version. We plan to gather more data and see if that improves accuracy in the CF assessment of the model.

5. Discussion and Future Work

Although even simple games like Prime Climb are extremely motivating for students, as we observed during our studies, there is currently very little evidence that simple or complex edu-games trigger learning. Usually this is not because of poor design, but because it is difficult to introduce intervention elements that make students reflect on domain knowledge without interfering with engagement. An accurate model of student learning is essential for balancing the trade-off between fostering learning and engagement in an educational game.

In this paper, we presented research to improve a model of student learning during the interaction with Prime Climb, an edu-game for number factorization. The model is to be used by a pedagogical agent that generates tailored interventions to trigger student reasoning when the student seems not to be learning well from the game. We discussed how we substantially improved the accuracy of an initial model by (i) changing the causality of the dependencies between knowledge and evidence nodes; (ii) learning model parameters from data. We also described a third version of the model that includes a common factor node to increase the specificity of the didactic advice that the model can support.

The next step in this research is to explore whether we can further increase model accuracy by (1) obtaining data to refine the part of the model that includes information on usage of the Magnifying Glass [3]; (2) including in the model the Prime Climb agent's interventions, which are currently not considered because we wanted to ascertain model accuracy before adding agent actions that relied on the model.

We also plan to run ablation studies to verify what impact the model accuracy has on overall effectiveness of the pedagogical agent. Finally, we wish to explore the scalability of our approach to modelling learning in more complex games and skills.

Acknowledgments

This research has been sponsored by an NSERC PGS-M scholarship. We thank Heather Maclaren helping with the user study, and Giuseppe Carenini for his help with data analysis

References

[1] Beck, J., P Jia and J. Mostow. Assessing Student Proficiency in a Reading Tutor That Listens. User Modeling 2003: pp. 323-327.

[2] Conati, C. and H. Maclaren. Data-driven Refinement of a Probabilistic Model of User Affect. To appear in User Modeling 2005.

[3] Conati, C. and X. Zhao. Building and Evaluating an Intelligent Pedagogical Agent to Improve the Effectiveness of an Educational Game. Intelligent User Interfaces 2004. pp. 6-13.

[4] Croteau, E. A., N. T. Heffernan and K. R. Koedinger. Why Are Algebra Word Problems Difficult? Using Tutorial Log Files and the Power Law of Learning to Select the Best Fitting Cognitive Model. Intelligent Tutoring Systems 2004. pp. 240-250.

[5] Klawe, M. When Does The Use Of Computer Games And Other Interactive Multimedia Software Help Students Learn Mathematics? NCTM Standards 2000 Technology Conference, 1998.

[6] Leemkuil, H., T. De Jong, R. deHoog, and N. Christoph. KM Quest: A collaborative Internet-based simulation game. Simulation & Gaming, 2003, 34(1).

[7] M. Mayo and A. Mitrovic. Optimising ITS Behaviour with Bayesian Networks and Decision Theory. International Journal of Artificial Intelligence in Education 2001. 12, pp 124-153.

[8] Nicholson, A.E., T. Boneh, T.A. Wilkin, K. Stacey, L. Sonenberg, V. Steinle: A Case Study in Knowledge Discovery and Elicitation in an Intelligent Tutoring Application. Uncertainty in Artificial Intelligence 2001.

[9] Randel, J.M., B.A. Morris, C.D. Wetzel, and B.V. Whitehill, The effectiveness of games for educational purposes: A review of recent research. Simulation & Gaming, 1992, 23(3).

[10] Schafer, R. And T. Weyrath. Assessing Temporally Variable User Properties with Dynamic Bayesian Networks. User Modeling 1997.

[11] VanLehn, K. Student modeling. Foundations of Intelligent Tutoring Systems. M. Polson and J. Richardson. Hillsdale, NJ, Lawrence Erlbaum Associates. (1988). pp. 55-78.

[12] VanLehn, K. and Z. Niu Bayesian student modeling, user interfaces and feedback: A sensitivity analysis. International Journal of Artificial Intelligence in Education, 2001. 12, pp. 154-184.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

On Using Learning Curves to Evaluate ITS

Brent Martin[1], Kenneth R Koedinger[2], Antonija Mitrovic[1] and Santosh Mathan[2]

[1]*Intelligent Computer Tutoring Group, University of Canterbury,*
Private Bag 4800, Christchurch, New Zealand
{brent,tanja}@cosc.canterbury.ac.nz

[2]*HCI Institute, Carnegie Mellon University, Pittsburgh, PA 15213*

Abstract. Measuring the efficacy of ITS can be hard because there are many confounding factors: short, well-isolated studies suffer from insufficient interaction with the system, while longer studies may be affected by the students' other learning activities. Coarse measurements such as pre- and post-testing are often inconclusive. Learning curves are an alternative tool: slope and fit of learning curves show the rate at which the student learns, and reveal how well the system model fits what the student is learning. The downside is that they are extremely sensitive to changes in the system's setup, which arguably makes them useless for comparing different tutors. We describe these problems in detail and our experiences with them. We also suggest some other ways of using learning curves that may be more useful for making such comparisons.

1 Introduction

Analysing adaptive educational systems such as Intelligent Tutoring Systems (ITS) is hard because the students' interaction with the system is but one small facet of their education experience. Pre- and post-test comparisons provide a rigorous means of comparing two systems, but they require large numbers of students and a sufficiently long learning period. The latter confounds the results unless it can be guaranteed that the students do not undertake any relevant learning outside the system being measured. Further, such experiments can only make comparisons at a high level: when fine-tuning parts of an educational system (such as the domain model), a large number of studies may need to be performed. In our research we have explored using a more objective measure of domain model performance, namely learning curves, to see if we can predict what changes could be made to improve student performance, including at the level of individual rules, or sets of rules. This often involves comparing disparate systems. In particular, we are interested in methods for comparing systems that work for small, short studies, so that we can propose, implement, test and refine improvements to our systems as rapidly as possible to make them maximally effective. The use of learning curves appears attractive in this regard.

Researchers use numerous methods to try to evaluate educational systems. Pre- and post-testing is commonly tried, but the results are often inconclusive. Often other differences are found in how students interacted with the system, but they appear to have been too little to give a clear test outcome. Ainsworth [1] failed to find significant pre-/post-test differences between REDEEM and CBT, but did find differences in certain situations. Similarly, Uresti and duBoulay [8] use pre-/post-testing to determine the efficacy of their learner companion across a variety of variables. They find no significant difference in learning outcome, but do find differences in measurements of usage within the tool.

Suraweera and Mitrovic [7] found significant differences between using their ITS (KERMIT) versus no tutor.

Because of the lack of clear results, researchers often measure other aspects of their systems to try to find differences in behaviour. However, these do not always measure learning performance specifically. Uresti and duBoulay measured the amount their "learning companion" was taught by the student during the session, which is arguably (but not explicitly) linked to improved learning. Walker et al [9] performed post-hoc analysis of the predictive ability of their collaborative information filter (which measures how well it chooses material), but they do not measure the effect on learning. Zapata and Greer [10] evaluated their inspectable Bayesian student modelling method by observation of the actions students performed and their interactions with the system, but again this does not measure changes in learning performance. Finally, many studies include the use of questionnaires to analyse student attitudes towards the system.

The use of learning curves attempts to bridge this gap by measuring learning activity within the system. As well as showing how well a particular system supports learning, they have the potential to allow quantitative comparisons between disparate systems. However, there are problems with such comparisons that need to be overcome. It is hoped that a better understanding of these curves and their limitations will add to the range of evaluative tools at our disposal.

Section 2 describes the use of learning curves for measuring ITS performance. We then describe the specific problems with comparing systems in Section 3, and examine some possible solutions, followed by a discussion in Section 4. Finally, we present our conclusions in Section 5.

2 Learning Curves

Learning curves plot the performance of students with respect to some measure of their ability over time. In the case of ITS, the standard approach is to measure the proportion of knowledge elements in the domain model applied by the student that have been used incorrectly, or the "error rate". Alternatives exist, such as the number of attempts taken to correct a particular type of error. Time is generally represented by the number of occasions the knowledge element has been used. This in turn may be determined in a variety of ways: for example, it may represent each *new problem* the student attempted that was relevant to this knowledge element, on the grounds that repeated attempts within a single problem are benefiting from the user having been given feedback about that particular circumstance, hence they may improve from one attempt to the next by simply carrying out the suggestions in the feedback without learning from them. If the student is learning the knowledge elements being measured, the learning curve will follow a so-called "power law of practise" [6]. Evidence of such a curve indicates that the student is learning the knowledge elements, or, conversely, that the elements represent what the student is learning: a poor power law fit suggests a deficient domain model. Therefore, when comparing two models we might argue that the model showing better power law fit is somehow superior.

The formula for a power law is:

$$Y = Ax^{-B} \quad (1)$$

The constant *A* represents the Y axis intercept, which for learning curves is the error rate at x=1, or the error rate prior to any practise. *B* depicts the power law slope, equivalent to the

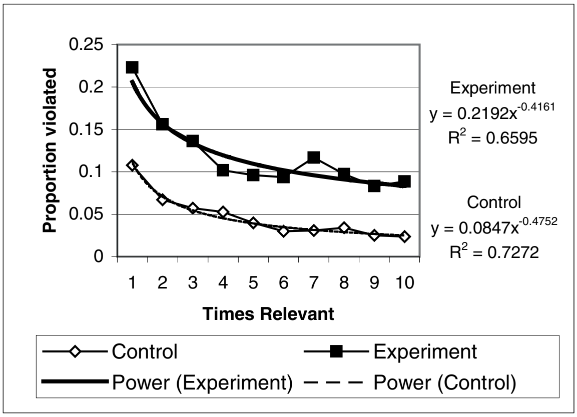


Fig. 1. Learning curves for two variants of SQL-Tutor

linear slope when the data is plotted using a log-log axis. This indicates the steepness of the curve, and hence the speed with which the student is learning the material. Finally, the fit of the power law to the data is measured. All of these may be used to compare two different approaches to determine which is better.

Data for learning curves is usually obtained post-hoc from student logs. For each student, a trace is generated for each knowledge element indicating the degree to which the student has correctly applied it. This may be a continuous value or simply “satisfied” or “violated”. Data values for a single knowledge element for a single student are unlikely to produce a smooth power law; they simply represent too little data. However, the data can be aggregated in several ways to represent useful summaries: data can be grouped for all students by knowledge element (to compare individual elements for efficacy), by student over all elements (to compare students) or over both for comparing different systems (e.g. two different domain models). The power law fit and slopes can then be compared. Fig. 1. illustrates this: the two curves represent the learning histories for two populations using different variants of the same ITS (SQL-Tutor [5]). The curve has been limited to the first 10 problems for which each constraint is relevant. This is necessary because aggregated learning curves degrade over time because the number of averaged data points decreases. Both curves exhibit a similar degree of fit, and their exponential slopes are similar. However, the Y asymptotes are markedly different, with the experimental group exhibiting more than double the initial error rate of the control group.

3 Problems with Comparing Models

Whilst it appears that learning curves can be compared with one another, there are several issues that call this practise into question. When comparing two different domain models, the power law parameters of fit and slope may be affected by incidental differences that arguably do not affect the quality of the model. These are now explored.

3.1 Fit versus Data Size

The quality of a power law tends to increase with data set size. A larger domain model is therefore likely to exhibit a better fit than a smaller one, even if it does not teach the student any better. For example, Koedinger and Mathan [3] compared learning outcomes associated with two types of feedback in the context of a spreadsheet tutor (an example of a cognitive tutor [2]). In the *Expert* version of the tutor, students were given corrective feedback as soon as they deviated from an efficient solution path. In the *Intelligent Novice* version, students were allowed to make errors; feedback was structured to guide students through error detection and correction activities. A learning curve analysis was performed to determine whether students in one condition acquired knowledge in a form that would generalize more broadly across problems. The tutor provided opportunities to practice six types of problems. A shallow mastery of the domain would result in the acquisition of a unique rule for each type of problem. A deeper understanding of domain principles would allow students to see the common abstract structure in problems that may seem superficially different. Consequently, students would acquire a smaller set of rules that would generalize across multiple problems. In the case of the spreadsheet tutor it was possible to use a set of four rules to solve the six types of problems represented in the tutor.

Two plots were created (Fig. 2), each with a different assumption about the underlying encoding. One plot assumed a unique rule associated with each of the six types of problems represented in the tutor. Thus, with each iteration through the six types of problems, there was a single opportunity to apply each production rule. In contrast, with a four skill, deep encoding, there were multiple opportunities to practice production rules that generalize across problems. Fitting power law curves to data plotted with these alternative assumptions about the underlying skill encoding might determine whether or not students were acquiring a skill encoding that would generalize well across problems.

Both graphs strongly suggest that the "intelligent novice" system is considerably better than the "expert" version – both fit and slope are considerably higher for this variant. However, the difference between the six- and four-skill models is not so clear. For both the expert and novice systems, the slope is higher for the four-skill model, suggesting more learning took place: this is particularly true for the "expert" system. However, in both cases the fit *decreases*, and again this is more marked in the "expert" system. At first glance these observations appear contradictory: learning is improved but quality of the model (as defined by fit) is lower. However, the four-skill model has 33% fewer knowledge elements than the original model, so we would expect the fit to degrade. This means we are unable to make comparisons based on fit in this case. Further, the comparisons of slope now arguably also become dubious. This latter concern could be overcome by plotting individual student curves and testing for a statistically significant difference in the average slopes, as described in Section 3.2.

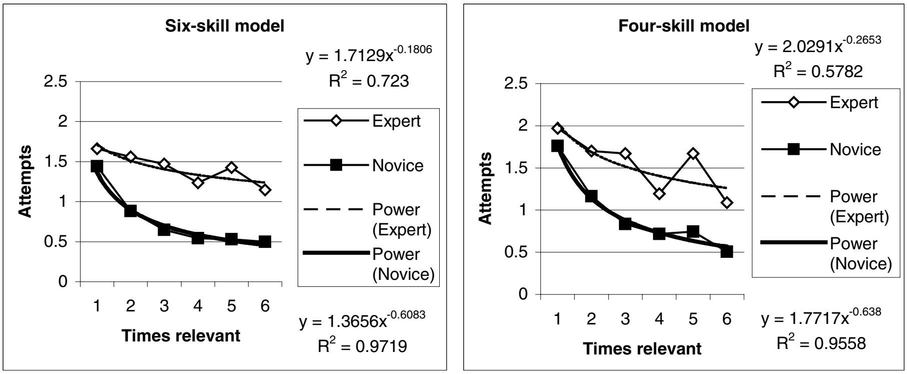


Fig. 2. Learning curves for six- versus four-skill models of the Excel tutor.

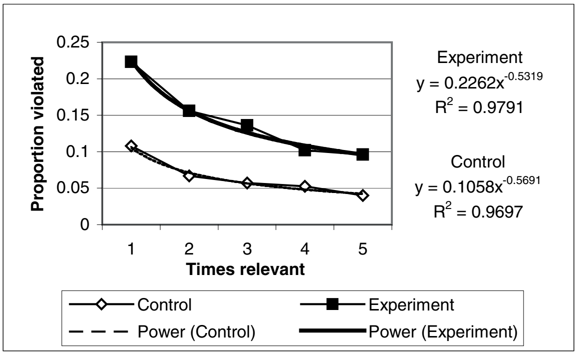


Fig. 3. Two variants of SQL-Tutor with different domain models

3.2 Initial versus Exponential slope

A serious issue with the use of power law slope is that it is highly sensitive to changes in the other parameters of the curve, particularly the Y axis intercept. In [4], we compared two versions of SQL-Tutor that had different problem sets and selection strategies. Fig. 3 shows the learning curves for the two systems trialled on samples of 12 (control) and 14 (experiment) University students. The two curves have similar fit and slope, which might lead us to conclude there is little difference in performance. However, the raw reduction in error suggests otherwise: between x=1 and x=5, the experimental group have reduced their error rate by 0.12, whereas the control group has only improved by 0.7, or about half.

The problem is that power law slope is affected by scale. Fig. 4 illustrates what happens if we modify the scale of a curve by multiplying each data point by two. Although this now represents twice the error reduction over time, the exponential slope is virtually unchanged. Further, adding a constant to the same data *reduces* the exponential slope considerably, even though the net learning is the same. In the case of our study, we were measuring differences caused by an improved problem selection strategy: if the new strategy is better, it should cause the student to learn a greater volume of new concepts at a time. The power law slope does not measure this. However, the Y axis intercept *does* reflect this difference, because it measures the size of the initial error rate. We argued therefore that by comparing the slope of the curve at x=1, we are measuring the reduction in error at the beginning of the curve, which represents how much the student is learning in absolute terms. For the graphs

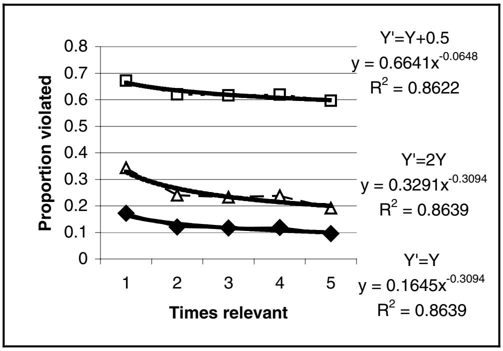


Fig. 4. Scale effects on learning curve slope

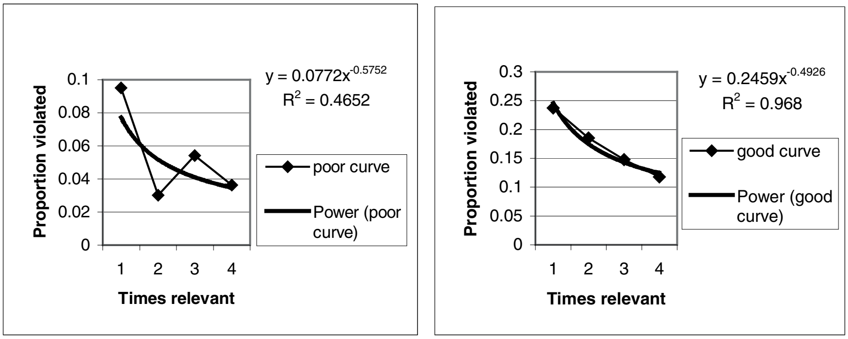


Fig. 5. Examples of individual student learning curves

in Fig. 4 this gives initial slopes of 0.12 for the experimental group and 0.06 for the control group, which correlates with the overall gain for x=5. The advantage of using initial slope rather than simply calculating the gain directly is that the former is using the best fit curve, which averages out errors across the graph, while the latter is a point calculation and is therefore more sensitive to error.

The fact that we have averaged the results across both all knowledge elements and students (in a sample group) may raise questions about the importance of the result. This is measured by plotting curves for individual students, calculating the learning rates and comparing the means for the two populations using an independent samples T-test. Fig. 5 shows examples of individual student curves. In general the quality of curves is poor because of the low volume of data, although some students exhibit high-quality curves. We have noticed a positive correlation between curve fit and slope. For the experiment described this yielded similar results to the averaged curves (initial learning rate = 0.16 for the experimental group and 0.07 for the control group). Further, the T-test indicated that this result was significant ($p<0.01$). We can therefore be confident that the experimental group exhibited faster learning of the domain model.

3.3 Early versus absolute learning

When evaluating learning curves, we assume that the power law of practise holds, and that the students' error rate will therefore trend towards zero errors in a negative exponential curve. However, there are arguably *two power laws superimposed*: the first is caused by simple practice, and should eventually trend to zero, although this may take a very long time. The second is caused by the feedback the system is giving: as long as this feedback is effective the student will improve, probably following a power law. However, we do not know how the effect of the feedback will vary with time: if it becomes less effective, the overall curve will "flatten", and thus deviate from a power curve. Even if the effect of feedback is constant (and therefore a curve based on feedback effect but not practice effect would trend to zero,) this curve may trend downwards much faster than the practice curve, and so will eventually intersect, and then be swamped by, the practise curve. The overall graph will therefore appear to be a power law trending to a Y asymptote greater than 0.

Fig. 6 illustrates this point. In this study, we compared two different types of feedback in SQL-Tutor on samples of 23 (control) and 24 (experiment) second year University students. The control system presented the student with the standard (low-level) feedback, while the

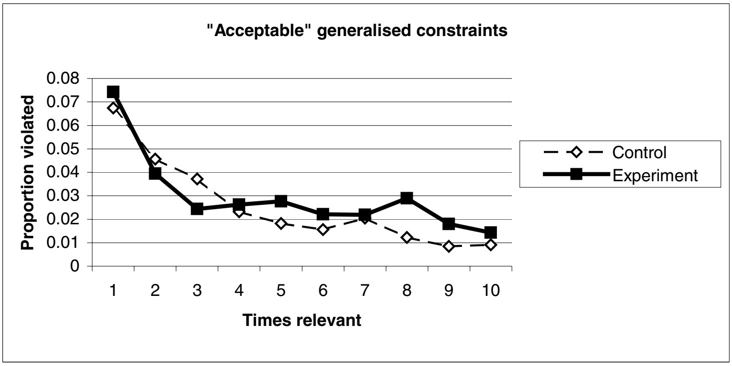


Fig. 6. Comparison domain models with differing feedback granularity

experimental system grouped several related knowledge elements together, and gave feedback at a more abstract level.

Over the length of the curves the amount of learning appears comparable between the two systems. However, the absolute gain for the *first two times the feedback was given* (i.e. the difference in Y between x=1 and x=3) is different for the two systems: For the control group the gain is around 0.03, while for the experimental group it is 0.05. We also notice that the curve for the experimental group appears to abruptly flatten off after this, suggesting that the feedback is only effective for the first two times it is viewed; after that it no longer helps the student.

We could use the initial learning rate again to measure the early gain, but this is unlikely to be useful because of the way the curve flattens off, and therefore deviates from the initial trend. (We could cut off the curve at x=3 but this is dubious since it is too few data points.) In this case we used the raw improvement as described in the previous paragraph. We obtained learning curves for individual students and performed a T-test on the value of error(t=3)-error(t=1) for each student. The results were similar to those from the aggregated graphs (mean error reduction = 0.058 for the experimental group and 0.035 for the control group), and the difference was significant ($p<0.01$).

4 Discussion

Section 3 illustrates some of the problems with comparing disparate systems using learning curves. These difficulties can be summarised into two main obstacles. First, changing the knowledge units being measured can affect the learning curves, even if there is no difference in learning. Conversely, learning differences may be masked by incidental effects. Consider, for example, two domain models that are identical, except that one of them includes a large number of trivially satisfied rules. For example, these rules might be useful in a different population, but turn out to be already known by the current students. These will have the effect of reducing the measured error rate, which leads to an *increase* in the exponential slope of the learning curve when compared to the model lacking these concepts, even though there is no improvement in learning. Further, it could be argued that this model is *worse* in the context of the current population. This could be alleviated by measuring the raw number of errors rather than the proportion of applied concepts that were incorrectly used, but such a measure would then depend on the overall size of the two systems being comparable, to say nothing of the number of concepts being applied at any one time. Thus a bias would appear towards more coarse-grained models. What is needed is some sort of normalisation of the curves.

The second problem is that the curves depend on both the domain model and the problems being set, as illustrated in [4]: setting hard problems involving the appropriate concepts appears to lead to steeper curves. To compare two domain models *only* would therefore require that the exact same problems are set, but this raises the spectre of the sequence of questions being better suited to one or other model.

There is also the question of what should be measured. With respect to fig. 6, it could be argued that the early differences in the curves are a detail only, and that overall learning is worse for the experimental group. However, the ideal behaviour of an education system's feedback arguably does *not* follow a power law: in the perfect system, the students would learn all concepts perfectly after seeing the feedback *once*. Further, gains at any point in the curve indicate superior behaviour in a limited context. In our case, the results suggest we should use general feedback the first few times it is presented; if the student still has problems with a concept, we should switch to more specific feedback. This is an important finding that warrants further investigation.

5 Conclusions

We have shown that education systems can be compared by using learning curves to measure the speed with which students learn the underlying domain model. However, if the systems being compared have different domain models, such comparisons are fraught with problems because of scaling effects; some means of normalising the curves is necessary if such comparisons are to be valid. Until this happens they should be presented with caution and treated with some scepticism. However, if the domain model is the same in the two systems, they can be directly compared.

Finally, we have not presented any empirical evidence that effects measured in learning curves translate into real differences in learning. Comparative studies using both learning curves and pre-/post-testing are needed to establish the relationship between learning curves and actual learning performance.

References

[1] Ainsworth, S.E. and Grimshaw, S., *Evaluating the REDEEM Authoring Tool: Can Teachers Create Effective Learning Environments?* International Journal of Artificial Intelligence in Education, 2004. 14(3): p. 279-312.

[2] Anderson, J.R., Corbett, A.T., Koedinger, K.R., and Pelletier, R., *Cognitive Tutors: Lessons Learned.* Journal of the Learning Sciences, 1995. 4(2): p. 167-207.

[3] Koedinger, K.R. and Mathan, S. *Distinguishing qualitatively different kinds of learning using log files and learning curves.* in *ITS 2004 Log Analysis Workshop*. 2004. Maceio, Brazil. p. 39-46.

[4] Martin, B. and Mitrovic, A. *Automatic Problem Generation in Constraint-Based Tutors.* in *Sixth International Conference on Intelligent Tutoring Systems*. 2002. Biarritz: Springer. p. 388-398.

[5] Mitrovic, A. and Ohlsson, S., *Evaluation of a Constraint-Based Tutor for a Database Language.* International Journal of Artificial Intelligence in Education, 1999. 10: p. 238-256.

[6] Newell, A. and Rosenbloom, P.S., *Mechanisms of skill acquisition and the law of practice*, in *Cognitive skills and their acquisition*, J.R. Anderson, Editor. 1981, Lawrence Erlbaum Associates: Hillsdale, NJ. p. 1-56.

[7] Suraweera, P. and Mitrovic, A., *An Intelligent Tutoring System for Entity RelationshipModelling.* International Journal of Artificial Intelligence in Education, 2004. 14(3): p. 375-417.

[8] Uresti, J. and Du Boulay, B., *Expertise, Motivation and Teaching in Learning Companion Systems.* International Journal of Artificial Intelligence in Education, 2004. 14: p. 67-106.

[9] Walker, A., Recker, M., Lawless, K., and Wiley, D., *Collaborative Information Filtering: a review and an educational application.* International Journal of Artificial Intelligence in Education, 2004. 14(1): p. 3-28.

[10] Zapata-Rivera, J.D. and Greer, J.E., *Interacting with Inspectable Bayesian Student Models.* Artificial Intelligence in Education, 2004. 14(2): p. 127-163.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

The role of learning goals in the design of ILEs: Some issues to consider[1]

Erika MARTÍNEZ-MIRÓN, Amanda HARRIS, Benedict DU BOULAY,
Rosemary LUCKIN and Nicola YUILL
IDEAs Lab, Departments of Informatics and Psychology, University of Sussex

Abstract. Part of the motivation behind the evolution of learning environments is the idea of providing students with individualized instructional strategies that allow them to learn as much as possible. It has been suggested that the goals an individual holds create a framework or orientation from which they react and respond to events. There is a large evidence-based literature which supports the notion of mastery and performance approaches to learning and which identifies distinct behavioural patterns associated with each. However, it remains unclear how these orientations manifest themselves within the individual: an important question to address when applying goal theory to the development of a goal-sensitive learner model. This paper exposes some of these issues by describing two empirical studies. They approach the subject from different perspectives, one from the implementation of an affective computing system and the other a classroom-based study, have both encountered the same empirical and theoretical problems: the dispositional/situational aspect and the dimensionality of goal orientation.

Keywords. learner modelling, goal orientation, motivation

1. Introduction

The AIED community has achieved considerable success in the development of software that can adapt to learners' needs whether they are working as individuals or in groups. To some extent these software systems emulate aspects of the role of a skilled teacher and improve learners' educational experience. Much of the work has focused on issues such as the representation of domain knowledge, human-computer interaction, and some aspects of teaching strategies (see [1] for a review). Although it is largely recognized that the learning process is greatly affected by the emotional and motivational state of the individual learner, it is only relatively recently that these issues have also been addressed. We are making progress towards an increased understanding of how an individual's cognitive and emotional states interact with each other and how this can help us to develop better intelligent learning environments (ILEs); systems that can recognize, acknowledge, and respond to emotional states by using, for instance, motivational tutorial tactics to promote learner affective states that are conducive to learning (e.g. [2]). In this

[1] This research was partially supported by Mexico's National Council of Science and Technology and the UK's Engineering and Physical Science Research Council

paper we explore the learner's goal orientation and the impact this can have upon their learning.

We report two studies with a common approach to the evaluation of a learner's goal orientation, but a different motivation for wishing to make this assessment. The first study is concerned with developing software that can adapt to a learner's goal orientation, and the second explores the ways in which goal orientation impacts upon learner engagement with collaborative learning using software. This work is important to the AIED community: as we develop increasingly sophisticated approaches to software scaffolding that address metacognitive and help-seeking behaviour (e.g. [3]), we also need to understand the influence of goal orientation. Similarly, work that aims to develop computer supported collaborative learning solutions will be informed by a greater understanding of the extent to which goal orientation interacts with a learner's collaborative style. At the heart of this is a need for us to understand more about what goal orientation is.

Achievement goal theory argues that the goals an individual pursues in an achievement context create a framework, or orientation, from which that individual interprets and reacts to subsequent events. These goals mediate internal processes and external actions and are important contributors to the self-regulatory processes involved in learning [4]. Examining the achievement goals a learner holds, therefore, informs our understanding of how individuals behave in learning contexts; vital information in the design of adaptive learning environments.

Two distinct orientations or patterns of achievement goals have been identified. An individual with a *performance goal orientation* interprets success as a reflection of their ability, they strive to receive positive judgments of their competence and avoid negative ones. In other words, they regard learning as a vehicle to public recognition rather than as a goal in itself. Somebody with a *mastery goal orientation*, in contrast, regards success as developing new skills, understanding content, and making individual progress: that is, learning is the goal itself.

These different learning goal orientations are associated with distinct behavioural patterns and learning strategies [4,5]. If a system can respond to the motivational orientation of individual learners, something expected of a human teacher, a more adaptive approach to learning may be encouraged, either by emphasizing a mastery approach by the tutor or by responding to the individual's own learning goal orientation. Further research needs to investigate the extent to which goals impact on the way in which learners interact with a computer system. We believe that having a better understanding of how individuals feel and act when interacting with a system could help with the ultimate goal of intelligent tutoring systems (ITSs) in customizing instruction for different student populations by, for instance, individualizing the presentation and assessment of the content. Exploring achievement goals may therefore be an important aspect of designing and constructing a learner model. However, we argue that if it is to be applicable in everyday educational contexts further empirical investigation into the nature of learning goals is needed. The following two empirical studies have highlighted the questions which remain unanswered within achievement goal theory and which, we argue, contribute to it being problematic, in its current form, when applied to specific educational contexts.

2. Individual differences in goal orientation

We describe two studies that address the individual differences that exist when different learners engage in the same task and the differential learning consequences of these differences. Both studies frame their investigation within an achievement goal perspective. Finally, they both use a standard method of measuring learning goals; the Patterns of Adaptive Learning Scales questionnaire (PALS) [6].

The first study looked at the way children interacted with two versions of an interactive learning environment that tried to emphasize a particular goal orientation by means of the feedback provided and some elements of the interface. The second study explored how goal orientations influence the way in which learners engaged in a computer-mediated collaborative task.

2.1. Study 1: Motivation and the influence of achievement goals

In recent years, modelling the student's motivational state has become a more recognised aspect in the design of interactive learning environments [7]. The current study investigated the role of students' goal orientations when interacting with educational software, in order to inform the design of more effective affective computing. The aim was to investigate, within a computer context, whether 1) emphasizing a particular goal orientation has an effect on individuals' performance; 2) a specific goal-oriented context works better for individuals according to their ability level; 3) an individual's goal orientation is overriden when they interact with a context that emphasizes a different goal orientation.

2.1.1. Method

A sample of 33 students, 9 to 11 years old, were asked to complete 1) a pre-test to assess their knowledge of the domain of ecology and 2) the PALS questionnaire [6] to assess their goal orientation. Then, they were allocated randomly to interact either with a mastery-oriented, performance-oriented or original version of the Ecolab (described below). A post-test was completed after the interaction with the system and a delayed post-test three weeks later.

2.1.2. The three different versions of Ecolab

The Ecolab II [8] is a system which was implemented within a Vygotskian design framework for the domain of ecology concepts such as food chains and food webs. The Ecolab plays the role of a more able partner that models how well the learner is doing and provides assistance accordingly. The Ecolab II was modified in order to implement two versions, one emphasizing a mastery goal orientation and the other a performance goal orientation [9]. Each version chooses an appropriate feedback strategy aimed to keep the student in a positive motivational state. For instance, if a student's persistence is low, her confidence is high and she has made an error, then the feedback provided promotes more persistence. In this case, the mastery system's motivational feedback might be "Learning how to do it requires another attempt", whereas the performance feedback might say "If you want to be the best, try again", in order to emphasize comparative judgements with other students. Along with the differences in motivational feedback, help is provided on demand in the mastery version, whereas the performance version offers help every time an incorrect action is performed (see [9]). In addition, elements of the interface are used to emphasize a particular goal orientation.

2.1.3. Results

When looking at cognitive strategies, e.g. help-seeking behaviour, or motivational strategies, e.g. expenditure of effort, no significant correlation with students' goal orientation or system used was found. When help was offered on demand, the students rarely made use of it, whereas in the case of automatic help the students did not have the choice of whether to accept it or not. In the light of these results, another study has been carried out, using adjusted versions of the software and increasing the interaction time with them, the analysis of the data is currently taking place. An important aim is to get empirical evidence to support or refute the claims that have been raised in achievement goal theory, particularly when considering a human-computer context.

2.2. Study 2: Collaborative learning and the influence of achievement goals

The results of Study 1 highlight some of the difficulties of applying achievement goal theory to the design of a single-user task. However, in school learning contexts, particularly during computer-mediated work, students will often work collaboratively. This raises additional questions about how to apply achievement goal theory to the design of a collaborative system, in which the goal orientation of not one but two learners will be important. In addressing this question, Study 2 explored the extent to which a child's goal orientation influences the way in which they interact and collaborate with a peer. This was a classroom-based study, in which pairs of students interacted with a non-intelligent system, but many of the same problems encountered in Study 1 became evident. This study, therefore, raises similar questions about our current understanding of learning goals, how they manifest themselves within the learner and how they are best applied to ILEs.

2.2.1. Method

A sample of 22 students aged 7 to 9 were observed participating in three collaborative sessions using a piece of software designed to guide their exploration of language awareness in joking riddles [10]. The aim of the study was to assess the nature of each student's participation in the interaction and relate this to their learning goal orientation. Collaboration was measured by analysing the language used by individual students. A coding scheme was designed for this purpose which consisted of 18 subcategories each falling into one of the following 5 language categories: Metacognitive comments, positive regulatory comments, negative regulatory comments, task specific comments and other comments. Learning goals were measured with the use of a teacher-rated questionnaire adapted from the PALS [6].

2.2.2. Results

Results indicate that learning goal orientation was significantly related to specific categories of language falling within the positive regulatory category. For example, the more mastery-oriented a child was, the more they engaged in constructive disagreements with their partner, $r = 0.62, p < 0.01$. On the other hand the more performance-oriented a child was, the less they engaged in this type of interaction, $r = -.413, p = 0.06$, a statistic approaching significance. A socio-constructivist approach to learning argues that in order for development to occur in the course of social interaction, students need to be able to resolve initially different perspectives in order to reach a new and joint understanding of the task at hand [11]. The results of this study indicate that the performance-oriented child may find this aspect of collaboration more difficult as they are less likely to vocalise disagreements than their mastery-oriented peers.

These results suggest a relationship between collaborative style and learning goal orientation, an interaction with warrants further investigation if a system is to scaffold collaborative interaction between users in relation to their learning goal orientation. However, these results need careful consideration in relation to the method of measuring learning goals. A child's orientation was decided by a median split, but in fact, most scores fell close to the neutral point and few could be classified as an extreme of either orientation. This suggests that learning goal orientation may not be as straightforward as the literature implies and that a given individual may be oriented towards both mastery and performance goals. Both studies found this problem with the PALS questionnaire, which raises methodological and theoretical issues about the way in which learning goal orientations are understood and consequently measured.

3. Current limitations of achievement goal theory

3.1. Dimensionality

There is no clear consensus within the literature about how to understand the constructs underlying mastery and performance goal orientations. For example, many authors understand the mastery/performance distinction as the end points on a single bipolar dimension, with a strong mastery goal orientation at one end and a strong performance goal orientation at the other [5,4]. Within this framework an individual can either be mastery-oriented or performance-oriented to a greater or lesser degree but not both. The other way learning goals have been understood are as separate dimensions that are neither mutually exclusive nor contradictory, but independent (e.g. [12,13]). The general perception from goal theory research is that performance and mastery goal orientations are part of a single dimension. While this is a theoretical issue, it has important consequences for studying achievement goals in real world learning contexts, an issue highlighted by difficulties we encountered in measuring learning goal orientations in the current two studies.

The PALS questionnaire [6] adopts an independent dimensions approach to the measurement of learning goals. Both studies found a similar effect using this scale, in that it was difficult, if not impossible, to classify individuals with orientations of mastery, performance-approach or performance-avoidant, as many scored high (or low) on all 3 dimensions. This suggests that it is not only possible to hold both mastery and performance approach goals simultaneously but also performance avoidance goals. Midgley *et. al.* (2000) suggest the PALS questionnaire should be used more as an indication of an individual's achievement goal tendency and not as a means of classification into one orientation or another [6]. However, in our studies there only ever appeared very slight tendencies one way or the other, with most students being rated similarly on all three goal dimensions. These results question an independent dimension approach, because if measuring goals in this way can mean an individual can hold different goals to the same extent at the same time, it does not account for the different cognitive, affective and behavioural patterns observed and associated with different orientations.

An alternative method is a forced choice measure adopted by Dweck which involves giving participants the choice between one of two tasks [14]. Each of the tasks appeals either to a mastery orientation, emphasising a learning dimension, or a performance orientation, emphasising the potential for demonstrating existing knowledge. The choice made by the participant is then taken as the measure of their goal orientation. This approach adopts a dichotomous view of learning goals in that the individual can not choose both tasks and, therefore, can only be classified as either performance- or mastery- oriented. While this solves the problems presented using the PALS questionnaire, i.e. one cannot be both orientations, it raises another, in that it does not assess the strength of an individual's goal orientation. It therefore forces participants into making the distinction, thereby pigeonholing them into one or other category without any opportunity to indicate the strength of their behavioural tendency. It also relies on making an inference from the behaviour displayed to the reason behind or motivation for that behaviour.

Neither of these approaches to the measurement of learning goal orientation takes into account the specific context in which a goal may be salient. The PALS questionnaire asks very broad questions about an individual's attitude toward learning, for example, "One of my goals is to show others that I'm good at my classwork." [6] (p.12). No reference is made to the specific type of classwork, the particular domain, or to whom the "others" refers, be they classmates, teachers or parents. In this sense the authors have attempted to keep each item on the questionnaire as context-free as possible. A similar attitude to context appears too in Dweck's task choice measure where she asks the participant whether they prefer "problems that aren't too hard" or "problems that I'm pretty good at" [14](p. 185).

Theorists have, therefore, deliberately attempted to decontextualise the way in which learning goal orientations are measured. However, it may be the very issue of context and how it influences the adoption of different learning goals that is fundamental to understanding the impact of learning goals on a learner's achievement behaviour. We argue this needs to be addressed if achievement goal theory is to have any practical use in the design and implementation of educational environments, computer supported or otherwise.

3.2. Dispositional vs. situational approach

The influence of context on learning goal orientation is related to the question of whether goal orientations can be considered as personality traits, stable across time and contexts, or as situational states which vary according to specific contexts. Goals are considered to be situational variables, when they are manipulated for the purposes of a given study (e.g. by means of task instructions [4], type of feedback [15], or retesting opportunities and criterion-referenced grading [16]). Studies which have attempted to do this have created mastery or performance contexts for short-term empirical measurements and have not followed up the extent to which goals have remained altered after experimental manipulation. The alternative perspective views goal orientation as stable and measurable dispositional traits. Studies adopting this perspective tend to measure the individual's orientation and how this influences their response patterns across situations (e.g. [12,17]).

Theorists adopt either a situational state or dispositional trait approach depending on their emphasis i.e. either developing classroom styles that are specifically designed to foster mastery goals [5,16] or understanding more about multiple goal perspectives before concluding that a mastery goal perspective is more adaptive [18]. Few have addressed the issue directly. However, it is our belief that this is another essential element in the understanding of learning goals and how they manifest themselves which needs more empirical evidence.

The resolution of this argument has implications for the way a system might use motivational dimensions to enhance a learning experience. For example, if goals are primarily dependent on context, regardless of an individual's goal orientation, then a context can be created to encourage the adoption of appropriate goals for that context. Alternatively, if the individual's orientation is stronger than environmental cues, learning activities can be designed to appeal to and match particular orientations. Taking this into account and considering the use of computer learning environments, a sensible approach to investigate how dispositional and situational variables interact within the individual is to design contexts that encourage the adoption of particular goals whilst also measuring the individual's dispositional traits. If a particular goal-oriented context proves to be "enough" to achieve a general improvement in learning, then it would be advisable to design learning activities according to that goal orientation. However, if more learning gains are found when individuals are exposed to goal-oriented contexts that match their goal orientation, then more attention needs to be focused on the simultaneous effects of both aspects: dispositional and situational.

4. Conclusions

The main goal in ITSs is to design systems that individualise the educational experience of students according to their level of knowledge and skill. Recent research suggests that their emotional state should also be considered when deciding the strategy to follow after an action has been taken.

This paper has focused on the importance of students' goal orientation. Achievement goal theory argues that different patterns of achievement behaviour become evident depending on the type of motivational orientation a learner adopts. However, we argue that further empirical investigation is needed, particularly as results from classroom-based studies question the way in which learning goal orientations and their impact are currently understood.

We argue particularly for the inclusion of context, such as a collaborative *vs.* an individual learning environment, to be considered an important variable in the understanding of learning goal orientations. This will have implications for the way in which learning goals are measured and defined. Current conflicting perspectives make it very difficult to measure learning goals and consequently their impact on students' behaviour in different contexts, which makes the application of achievement goal theory particularly difficult. We believe that exploring the role of context explicitly may go some way to resolving some of the current limitations. Future work will aim to identify ways of implementing a context-specific goal perspective in the design of ILEs.

References

[1] B. du Boulay and R. Luckin, "Modelling human teaching tactics and strategies for tutoring systems," *International Journal of Artificial Intelligence in Education*, vol. 12, no. 3, pp. 232–234, 2001.

[2] W. L. Johnson, S. Kole, E. Shaw, and H. Pain, "Socially intelligent learner-agent interaction tactics," in *Artificial Intelligence in Education* (J. K. Ulrich Hoppe, Felisa Verdejo, ed.), (Amsterdam), pp. 431–433, IOS Press, 2003.

[3] V. Aleven, B. McLaren, I. Roll, and K. Koedinger, "Toward tutoring help seeking: Applying cognitive modeling to meta-cognitive skills," in *7th International Conference on Intelligent Tutoring Systems, ITS 2004* (F. P. James C. Lester, Rosa Maria Vicari, ed.), (Berlin), pp. 227–239, Springer-Verlag, 2004.

[4] E. S. Elliot and C. S. Dweck, "Goals: An approach to motivation and achievement," *Journal of Personality and Social Psychology*, vol. 54, pp. 5–12, 1988.

[5] C. A. Ames, "Classrooms: Goals, structures, and student motivation," *Journal of Educational Psychology*, vol. 84, pp. 261–271, 1992.

[6] C. Midgley, M. Maehr, L. Hruda, and E. Anderman, *Manual for the Patterns of Adaptive Learning Scales (PALS).* Ann Arbor, MI: University of Michigan, 2000.

[7] C. Conati, "Probabilistic assessment of user's emotions in educational games," *Journal of Applied Artificial Intelligence*, vol. 16, no. Special issue on 'Merging Cognition and Affect in HCI', pp. 7–8, 2002.

[8] R. Luckin and B. du Boulay, "Ecolab: The development and evaluation of a Vygotskian design framework," *International Journal of Artificial Intelligence in Education*, vol. 10, pp. 198–220, 1999.

[9] E. A. Martínez-Mirón, B. du Boulay, and R. Luckin, "Goal achievement orientation in the design of an ILE," in *Workshop on Social and Emotional Intelligence in Learning Environments at the 7th International Conference on Intelligent Tutoring Systems*, (Maceio, Brazil), 2004.

[10] N. Yuill and J. Bradwell, "The laughing PC: How a software riddle package can help children's reading comprehension," in *Proceedings of the BPS Annual Conference*, (Brighton, UK), p. 119, 1998.

[11] A. Garton, *Social interaction and the Development of Language and Cognition*, ch. Social explanations of cognitive development. Psychology Press, 1992.

[12] A. Valle, R. G. Canabach, J. C. Nunez, J. Pienda, S. Rodriguez, and I. Pineiro, "Multiple goals, motivation and academic learning," *British Journal of Educational Psychology*, vol. 73, pp. 71–87, 2003.

[13] J. L. Meece and K. Holt, "A pattern analysis of students' achievement goals," *Journal of Educational Psychology*, vol. 85, no. 4, pp. 582–590, 1993.

[14] C. S. Dweck, *Self-theories. Their role in motivation, personality, and development.* Psychology Press, Taylor and Francis Group, 2000.

[15] R. Butler, "Task-involving and ego-involving properties of evaluation: Effects of different feedback conditions on motivational perceptions, interest, and performance," *Journal of Educational Psychology*, vol. 79, no. 4, pp. 474–482, 1987.

[16] M. V. Covington and C. L. Omelich, "Task-oriented versus competitive learning structures: motivational and performance consequences," *Journal of Educational Psychology*, vol. 78, no. 6, pp. 1038–1050, 1984.

[17] K. E. Barron and J. M. Harackiewicz, "Achievement goals and optimal motivation: testing multiple goal models," *Journal of Personality and Social Psychology*, vol. 80, pp. 706–722, 2001.

[18] J. M. Harackiewicz and A. J. Elliot, "The joint effects of target and purpose goals on intrinsic motivation: A mediational analysis," *Personality and Social Psychology Bulletin*, vol. 24, no. 7, pp. 675–688, 1998.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

A knowledge-based coach for reasoning about historical causation

Liz MASTERMAN
Learning Technologies Group, Oxford University, UK

Abstract. The ability to explain the causes of historical events is a key skill for learners to acquire, but the ill-structured nature of the task means they cannot be guided through a problem-space of well-defined moves to reach a correct answer. This paper investigates whether a knowledge-based computer coach can provide effective guidance to learners as they construct diagrammatic explanations of the causes leading to a particular event. The design of the coach was based on a model of expert reasoning synthesised from the historiographical literature and on an analysis of teacher-learner interactions observed during classroom activities. Coaching was provided at two levels: a) generalised (decontextualised) guidance and b) guidance directly relevant to the topic of study. Where appropriate, learners could choose to disregard the coach's advice. The knowledge-base underlying the coach could also be made available as a scaffolding aid. An evaluation with three groups of students aged 12-13 showed that i) maximal scaffolding and content-specific coaching resulted in diagrammatic explanations of greater accuracy and superior structural quality to those produced either with generalised guidance or with no guidance at all, and ii) learners' appreciation of the subjective nature of historical explanations was not compromised by the coaching interventions.

Introduction

Causation is one of a set of key concepts that provide both experts and learners with a structure for understanding and thinking about history [11]. However, reasoning about historical causation—that is, identifying and explaining the relationships between a particular event and those antecedent events that may be said to have brought it about—poses a substantially greater challenge than reasoning about causation in everyday life or in the natural sciences or law. The uniqueness of historical events and their remoteness in time mean that the historian can neither conduct experiments that make the effects of such events perceptible nor interrogate historical personages about their motives and intentions. Instead, he or she must rely on a vast knowledge-base of evidence derived from sources which may be incomplete, inconsistent and difficult to interpret. Thus, an answer to the question "why?" about history can never be definitive; rather, it is a matter of personal interpretation influenced in part by the historian's general knowledge and perspective.

A key challenge in fostering learners' reasoning about causation, therefore, is to induce them to appreciate the admissibility of alternative solutions to problems of historical causation. This paper outlines a study which investigated whether a knowledge-based coaching system can provide effective support for learners' emergent reasoning as they construct historical explanations on the computer. It begins by outlining the characteristics of expert reasoning about causation before examining how teachers introduce their learners to the task and scaffold the solving of "causation" problems through verbal interactions with learners and through different forms of external representation. The paper then describes how the information obtained from this research provided the basis for a computer program in which learners constructed diagrammatic explanations of the causes of the English Civil War. Finally, it reports and discusses the findings of an evaluation comparing the outcomes of learners' use of the program with and without support from a computer-based coach.

1. Characteristics of expert reasoning about historical causation

A model of expert performance in a particular domain can give an indication of what the outcome of successful learning should look like [1]. However, constructing such a model for historical causation is a challenging task, since there is neither an agreed terminology nor an agreed set of procedures among the experts, with historiographers arguing the case for and against causal reasoning as a deductive, inductive, adductive or associative process. It is, however, best characterised as an *informal* logic, governed by internal principles which have more to do with rhetoric than with propositions of formal logic [17] or estimates of probability. In order to identify those concepts and procedures most commonly associated with this logic, the author undertook an extensive synthesis of the historiographical literature on causation. Figure 1 summarises the outcome of this task.

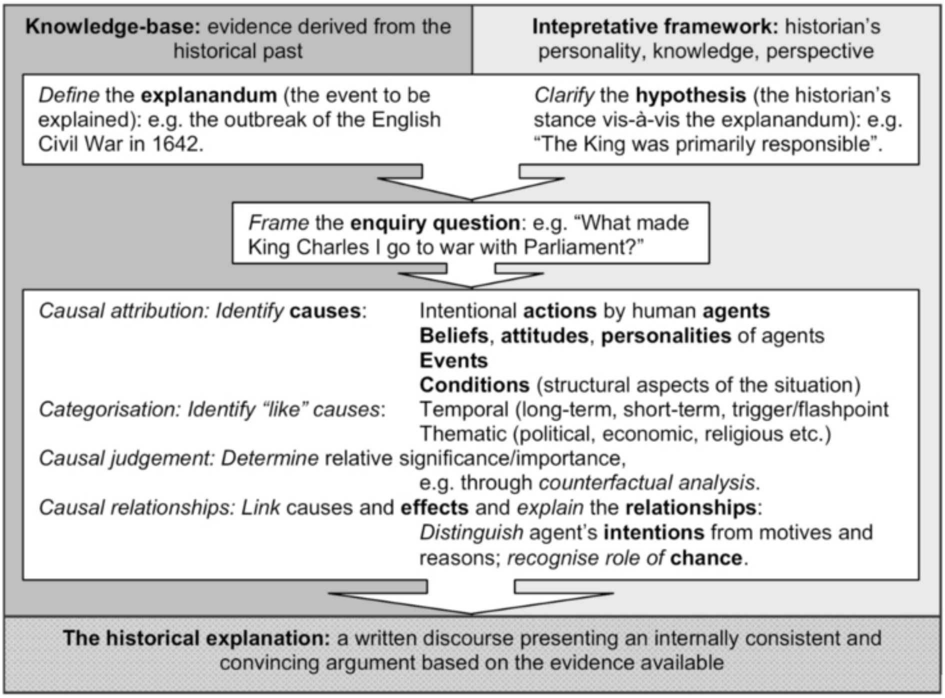

Knowledge-base: evidence derived from the historical past
Intepretative framework: historian's personality, knowledge, perspective
Define the explanandum (the event to be explained): e.g. the outbreak of the English Civil War in 1642.
Clarify the hypothesis (the historian's stance vis-à-vis the explanandum): e.g. "The King was primarily responsible".
Frame the enquiry question: e.g. "What made King Charles I go to war with Parliament?"
Causal attribution: Identify causes:
Intentional actions by human agents
Beliefs, attitudes, personalities of agents
Events
Conditions (structural aspects of the situation)
Categorisation: Identify "like" causes:
Temporal (long-term, short-term, trigger/flashpoint
Thematic (political, economic, religious etc.)
Causal judgement: Determine relative significance/importance, e.g. through counterfactual analysis.
Causal relationships: Link causes and effects and explain the relationships:
Distinguish agent's intentions from motives and reasons; recognise role of chance.
The historical explanation: a written discourse presenting an internally consistent and convincing argument based on the evidence available

Figure 1. Reasoning about historical causation: summary of the principal **concepts** and associated *procedures*. Synthesised from numerous sources cited in [9]

To arrive at a historical explanation, the historian applies his or her interpretative framework to the knowledge base of sources in order to identify the relevant causes and to categorise, and/or judge the significance of, these different causes as desired. Establishing a cause-effect relationship is relatively straightforward in the case of conditions and events, especially where the agent is a force of nature. However, determining causal relationships where human actions are involved necessitates inferring the agent's conscious intentions, as distinct from their motives (often unconscious) and reasons (how the agent might justify the action). It should be recognised, of course, that the procedures in Figure 1 are iterative rather than sequential; that is, an attempt to establish a causal relationship between two factors might trigger a return to the evidence to search for a third, intermediate, factor.

Perhaps the cognitive model to which reasoning about historical causation is closest is that for the solving of ill-structured problems [18]. In line with this class of problem, historical causation is distinguished by i) an initial state (the explanandum) and a goal state (the historical explanation), each of which may be open to multiple interpretations; ii) the presence of a large number of open constraints (i.e. gaps and inconsistencies in the evidence) which different members of the problem-solving community may fill in different ways, thereby leading to iii) differing solutions, the quality of which is largely a matter of pragmatic judgement. Furthermore, as with other ill-structured problems, constructing a historical explanation involves selecting relevant information from a considerable body of data and decomposing the main problem into multiple relatively well-structured problems.

The ramification of this model for history teachers is clear in that, unlike problem-solving tasks in maths, science or logic, they cannot direct learners through a problem space of well-defined moves where specific constraints must (and can) be satisfied in order to arrive at the "correct" answer. Indeed, they must actively avoid creating the impression that problems of historical causation are solved in this way.

2. Promoting reasoning about historical causation in the classroom

While providing clues to the nature of expert reasoning, historiographers give little guidance about how to guide learners towards the desired performance [19]. Therefore, in order to determine how far the model of reasoning presented in Figure 1 is reflected in the classroom, what sorts of misconceptions learners have, how teachers guide learners through the problem-solving task and what forms of representation they use to mediate this process, the author combined a survey of recent research on the development of learners' causal reasoning [e.g. 8] and a review of the literature on teaching causation in the UK [e.g. 6] with classroom observations. The observations covered 46 lessons on a range of "causation" topics, involving students aged from 11 to 17 in three mixed-ability co-educational schools and one school for medium- to high-ability girls. The aim was to establish, from these multiple sources of data, generalisations applicable to the design of the proposed program.

2.1 Introducing learners to the concepts and procedures involved in causal reasoning

There was no overt teaching of any "global" logic for reasoning about historical causation in any of the schools observed. Rather, concepts and procedures were introduced gradually, according to the demands of the subject matter and the teacher's perception of the students' readiness for tackling new concepts or familiar ones at a higher level. Nevertheless, the principal elements of Figure 1 were discernible in the observations, albeit in a somewhat simplified form. For example, students, with their initially naïve interpretative frameworks, were not expected to generate their own hypotheses and so were given enquiry questions which had been pre-defined by the teacher. Overall, therefore, teachers may be seen as fostering a model of *competent*, rather than expert, reasoning which students in the UK might be expected to acquire before they end their compulsory study of history at age 14.

The observations also validated the equation of reasoning about historical causation with the solving of ill-structured problems, in that teachers laid stress on the multiplicity of possible solutions; provided students with a subset of sources (usually from a textbook);

and subdivided the topic into manageable phases: information-gathering and interpretation, knowledge-construction (categorising, judging significance, identifying causal relationships) and knowledge communication (usually as a written historical explanation).

2.2 Teacher-learner interactions

The analysis of observed interactions revealed a high level of input by teachers during the information-gathering phase in helping students to interpret evidence, alerting them to their misconceptions and explaining archaic terms and the abstract concepts associated with historical causation. In the knowledge-construction phase, when students were engaged in semi-independent problem-solving activities, the teacher would move around the class and engage with students individually, quickly reviewing their work and offering advice and feedback (i.e. coaching). The observations yielded four *styles* of coaching intervention:

- *Directive:* unsolicited advice and hints at the outset of an activity.
- *Responsive:* guidance in response to a student's request for help.
- *Reactive:* immediate feedback on an action by an individual student.
- *Retrospective:* holistic feedback either to an individual learner or to the whole class when the activity has reached an advanced stage or has been completed.

2.3 Representations of knowledge

The outcome of a causation enquiry is normally a written explanation, an exercise which is fraught with difficulty for many learners [4]. Therefore, teachers tended to scaffold this process by helping students to formulate their ideas through constructing representations which made fewer demands on students' linguistic abilities; for example, tabulated charts, directed graphs and "cause cards" (that is, labelled slips of paper, each corresponding to a cause, which pupils sorted into different configurations in categorisation activities).

3. Design of the computer-based coach

The next stage in the study was to feed the findings from the observation into the design of the program, titled 20/20. This design hinged on three interrelated decisions: i) the phase(s) in a causation enquiry which the program would support; ii) the role of the computer coach vis-à-vis the teacher; and iii) the form of representation to be supported at the interface. These decisions were made by marrying observational data with a theoretical framework which places teacher-learner interactions and learning activities within a *modelling-supporting-fading* paradigm [9, 10], where the teacher adopts the role of more able partner. The observational data suggested that there would be almost insurmountable difficulties in implementing a system in which the computer assumed the role of *replacement* teacher, since teachers often used topical references or their personal knowledge of students when explaining abstract concepts. However, it was also noted that, during classroom activities, the teacher did not always have time to provide guidance to individual students. Hence, it seemed that the computer could fulfil the role of *adjunct* to the teacher by coaching students when the latter was unavailable. However, the teacher would remain responsible for diagnosing learners' levels of ability and deciding the amount of support to be provided by the computer.

The representational form, a diagram akin to a concept map, was chosen because of its simplicity (consisting of two basic elements: boxes and arrows) and because it combined two forms already used in the classroom: namely, directed graphs and cause cards. The guiding principle in devising the notation was the need for a perspicuous scheme which did not impose an additional cognitive burden on students and made it possible to represent multiple perspectives simultaneously (e.g. temporal classifications plus thematic groupings). Table 1 maps the key concepts associated with historical causation supported by 20/20 to the notation used. Figure 2 shows the notation in context: a student's diagram.

Table 1. The key concepts associated with causation and their representation at the interface

Concept	Examples	Notation used	Rationale for notation
Causal factor	Action, event, condition	Box with text label	Visual similarity to cause card
"Temporal" classification	Long-term, short-term, flashpoint/trigger	a) Dark cloud, yellow or red lightning flash *or* b) Binoculars, spectacles	a) Meteorological metaphor is associated with build-up to cataclysmic events e.g. wars. b) Ocular metaphor is more suited to undramatic events.
Thematic grouping	Political, economic, religious, military	Colour-coding in boxes	Cultural associations (where possible); e.g. red = military
Significance	Major cause, minor cause	Variations in thickness of box borders	Visual salience
Chance factor		**!** in cause box	Used on "hazard" road signs
Causal relationships	N/A	Arrow connecting cause to effect	Arrow is suggestive of causal stimulus

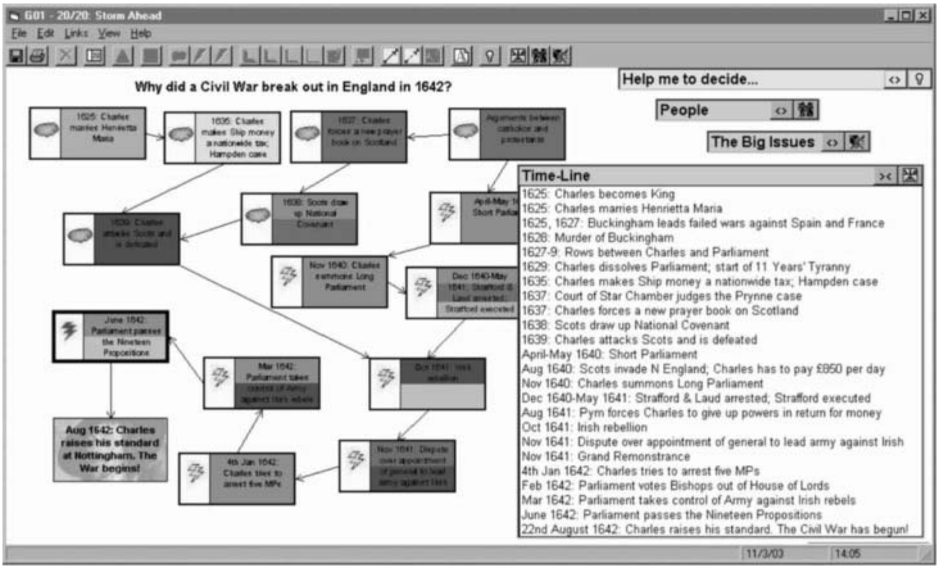
The core system consisted of a "workspace" where learners explored and experimented with their ideas, creating and manipulating configurations of cause boxes and links to build a diagrammatic representation of the causes of the event in question (see Figure 2).The procedures involved in causal reasoning were mostly carried out through "point-and-click" operations using buttons in the toolbar.

The central challenges in designing the coach which was to be overlaid on the core system were primarily pedagogical; viz. i) how to guide learners towards a plausible solution to the question while simultaneously reinforcing an appreciation of the subjective quality of that solution, and ii) how to diagnose the misconceptions behind their actions. To meet both challenges, moves that could prompt coaching interventions were divided into:

- *Strong issues:* illogical moves (e.g. linking an effect to its cause instead of vice versa), in which the coach would always intervene to enforce correction.
- *Weak issues:* matters that were open to interpretation. Here, the coach would display a pop-up message alerting the learner to the discrepancy between their diagram and its own view, but give the learner the freedom to ignore its advice.

The frequency of interventions was defined by a set of rules derived from the WEST system [2] and by experimentation. The style of interventions by the computer coach was determined both by observational data and by technological constraints. For example, the object-oriented behaviour of the interface (i.e. select object→perform action) precluded directive coaching for almost all moves. Also, to avoid processing natural-language input, responsive coaching was implemented as a list of frequently-asked questions under the heading "Help me to decide".

The design allowed for two levels of coaching (as well as none at all), with the teacher predetermining the level to be used with any one group of learners. "Generalised" coaching gave broad guidance only (e.g. decontextualised definitions of concepts). "Content-specific" coaching offered additional guidance relevant to the situation in question, although this meant restricting learners to choosing causes from three pre-defined lists: actions and events (the "Time-Line" in Figure 2), beliefs and attitudes of the agents involved ("People"), and the underlying conditions ("Big Issues"). These lists could also be made available as optional scaffolding aids for learners receiving generalised (or no) coaching.


G01 - 20/20: Storm Ahead
File Edit Links View Help
Why did a Civil War break out in England in 1642?
1625: Charles marries Henrietta Maria
1635: Charles makes Ship money a nationwide tax; Hampden case
1637: Charles forces a new prayer book on Scotland
1638: Scots draw up National Covenant
1639: Charles attacks Scots and is defeated
April-May 1640: Short Parliament
Nov 1640: Charles summons Long Parliament
Dec 1640-May 1641: Strafford & Laud arrested; Strafford executed
June 1642: Parliament passes the Nineteen Propositions
Mar 1642: Parliament takes control of Army against Irish rebels
Oct 1641: Irish rebellion
Aug 1642: Charles raises his standard at Nottingham. The War begins!
4th Jan 1642: Charles tries to arrest five MPs
Nov 1641: Dispute over appointment of general to lead army against Irish
Help me to decide...
People
The Big Issues
11/3/03
14:05

Time-Line

1625: Charles becomes King
1625: Charles marries Henrietta Maria
1625, 1627: Buckingham leads failed wars against Spain and France
1628: Murder of Buckingham
1627-9: Rows between Charles and Parliament
1629: Charles dissolves Parliament; start of 11 Years' Tyranny
1635: Charles makes Ship money a nationwide tax; Hampden case
1637: Court of Star Chamber judges the Prynne case
1637: Charles forces a new prayer book on Scotland
1638: Scots draw up National Covenant
1639: Charles attacks Scots and is defeated
April-May 1640: Short Parliament
Aug 1640: Scots invade N England; Charles has to pay £850 per day
Nov 1640: Charles summons Long Parliament
Dec 1640-May 1641: Strafford & Laud arrested; Strafford executed
Aug 1641: Pym forces Charles to give up powers in return for money
Oct 1641: Irish rebellion
Nov 1641: Dispute over appointment of general to lead army against Irish
Nov 1641: Grand Remonstrance
4th Jan 1642: Charles tries to arrest five MPs
Feb 1642: Parliament votes Bishops out of House of Lords
Mar 1642: Parliament takes control of Army against Irish rebels
June 1642: Parliament passes the Nineteen Propositions
22nd August 1642: Charles raises his standard. The Civil War has begun!

Figure 2. Workspace of 20/20, here subtitled "Storm Ahead" because the meteorological icons are in use. Two of the lists of pre-defined causes are closed, as is the set of issues for which responsive coaching is available

The coaching system was implemented as a combination of a) immutable rules embedded in the program code along with generalised coaching messages applicable to all situations, and b) a knowledge base of pre-defined causes specific to a particular historical situation, stored in a database along with their attributes (including relationships with other causes) and the coaching messages relating directly to them. This knowledge base thus served two purposes: to scaffold learners' tasks and to function as an "expert version" with which the coach could compare the outcome of learners' actions and give fully contextualised guidance.

4. Evaluation

The 20/20 coach was evaluated with three mixed-ability classes of students aged 12-13 at one of the co-educational schools involved in the observations. The hypothesis proposed that students who received higher levels of computer-based support would produce diagrams that were i) more accurate (i.e. closer to the expert version) and ii) of superior structural quality (i.e. containing more causes and links) than students who received less support. Each class constituted a separate experimental condition (see Table 2). They had already studied the causes of the English Civil War and spent two one-hour sessions using 20/20 to construct a diagram explaining why, in their view, the war broke out.

Table 2. Experimental conditions in the evaluation of the 20/20 coach

	Group T (26 students)	Group G (22 students)	Group N (20 students)
Composition	26 students; Teacher A	22 students; Teacher B	20 students; Teacher B
Scaffolding	Use causes from pre-defined lists only	Select causes from pre-defined lists + optionally devise their own causes from researching in their textbooks	
Coaching	Content-specific	Generalised	None

Analysis of the records of learners' actions in 20/20 confirmed that group T did receive more coaching: one reactive intervention per 5.33 actions and one retrospective intervention per 72.36 actions, compared with 28.14 and 65.56 for group G. Both groups appeared to act on the computer's coaching of "weak" issues roughly two-thirds of the time, suggesting that they were not completely in thrall to the computer coach. Recourse to responsive coaching was minimal, with a total of 21 requests from the two groups.

The completed diagrams of all three groups were scored using formulae based on [5, 13] and described in detail in [9]. Accuracy scores could range from 1 (maximum) down to 0 (minimum), and scores for structural quality could range from 1 (maximum) down to values below 0. Table 3 summarises the mean scores and the results of statistical tests performed on them.

Table 3. Mean scores (and standard deviations) obtained by the three groups, and results of statistical tests

Criterion	**Group T**	**Group G**	**Group N**	**Kruskal-Wallis test**
Accuracy: cause boxes	0.57 (0.15)	0.19 (0.13)	0.26 (0.09)	$\chi^2 = 25.386$, p = .000
Accuracy: links	0.03 (0.03)	0.01 (0.02)	0.02 (0.02)	$\chi^2 = 5.705$, p = .058
Structural quality	0.53 (0.04)	0.14 (0.04)	0.17 (0.28)	$\chi^2 = 9.605$, p = .008

Differences among the groups were significant at p<=.05 except for the accuracy of links, where the differences approached significance. However, it is notable not only that group T's scores were well ahead of the other two groups, but also that group G actually scored slightly lower than group N. Hence, the hypothesis was only partly supported.

5. Discussion

The question investigated in this paper is whether a knowledge-based coach can provide effective support for learners' emergent reasoning about historical causation. Since causal reasoning is a skill which requires several years to develop, it was possible to evaluate only a short-term intervention. Findings from the 20/20 evaluation showed that varying the amount and content of computer-based support could result in differences in performance in a single task without excessively compromising learners' independence of thought. However, it appeared that diagrams of significantly higher quality were produced only where a) the level of scaffolding was sufficiently high as to minimise the risk of students' voluntarily making unacceptable moves, and b) the coaching delivered was relevant to the topic of study. Coaching which offered only generalised advice and feedback often resulted in diagrams that differed little from those produced without any coaching at all—perhaps because such advice provided insufficient clues as to how learners should act in a specific situation. Although group T had a different teacher, observational notes from the evaluation sessions suggest that differences in the two teachers' styles were insufficient to account for such large variations in scores. Nevertheless, the investigation would benefit from a) a longitudinal study to determine, inter alia, whether learners can generalise from the advice received in relation to one historical situation and apply it, after an extended period, to a novel situation, and b) more rigorous control of variables such as teaching styles.

The program 20/20 is innovative in that it supports a domain traditionally under-represented in artificial education research, viz. history (an exception is Disciple [14]), but it also continues a well-established tradition of intelligent graphical reasoning tools that includes Belvedere, Convince Me and Reason!Able [15, 12, 16], as well as the more recent Reasonable Fallible Analyser (RFA) [3]. The option of a content-specific knowledge-based coach has commonalities with Belvedere; however, 20/20 does not currently support learners' construction of a substantiated argument like Belvedere, Convince Me and Reason!Able or allow learners to argue in favour of their position, as does the RFA. It would be worthwhile, therefore, to consider adding either or both of these facilities to 20/20.

Ultimately, further developments to the 20/20 coach must recognise the central tension between that which can be achieved technologically and that which is acceptable historiographically and, hence, pedagogically. At present, a major limitation of 20/20 is the lack of coaching for the key procedure of explaining causal relationships. Yet it is not only impossible to formulate the universal rules that might underlie a coach for this task (e.g. "people of disposition X faced with situations of type Y are likely to act in manner Z"), but such rules would negate the very essence of historical causation: namely, to explain why particular individuals acted as they did in specific situations [7]. History may be full of ill-structured problems with diverse solutions, but its internal logic must be strictly observed.

With acknowledgements to Mike Sharples for his invaluable support during the study.

References

[1] Bransford, J.D., Brown, A.L. & Cocking, R.R. (Eds.) (1999). *How People Learn: Brain, Mind, Experience, and School*. Washington, DC: National Academy Press.

[2] Burton, R.R. & Brown, J.S. (1982). An investigation of computer coaching for informal learning activities. In D. Sleeman & J.S. Brown (Eds.), *Intelligent Tutoring Systems* (pp. 79-98). London: Academic Press.

[3] Conlon, T. (2004). 'Please Argue, I Could Be Wrong': a Reasonable Fallible Analyser for Student Concept Maps. AACE Journal, 12(4). Available: http://dl.aace.org/15571 [Accessed 25/01/05]

[4] Curtis, S. (1994). Communication in History—A process based approach to developing writing skills. *Teaching History*, 77, 25-30.

[5] Funke, J. (1985). Steuerung dynamischer Systeme durch Aufbau und Anwendung subjectiver Kausalmodelle. *Zeitschrift fur Psychologie*, 193(4), 443-465.

[6] Husbands, C. (1996). *What is history teaching? Language, ideas and meaning in learning about the past*. Buckingham: Open University Press.

[7] Lee. P.J. (1984). Why Learn History? In A.K. Dickinson, P.J. Lee and P.J. Rogers (Eds.), *Learning History* (pp. 1-19). London: Heinemann.

[8] Lee, P., Dickinson, A. and Ashby, R. (1998). Researching Children's Ideas about History. In J.F. Voss and M. Carretero (Eds.), *Learning and Reasoning in History* (pp. 227-251). London: Woburn Press.

[9] Masterman, E.F. (2004). *Representation, mediation, conversation: integrating sociocultural and cognitive perspectives in the design of a learning technology artefact for reasoning about historical causation*. Unpublished doctoral thesis, University of Birmingham, UK.

[10] Masterman, L. and Sharples, M. (2002). A theory-informed framework for designing software to support reasoning about causation in history. *Computers & Education*, 38, 165-185.

[11] Nichol, J. (1999). Who wants to fight? Who wants to flee? Teaching history from a "thinking skills" perspective. *Teaching History*, 95, 6-13.

[12] Schank, P. & Ranney, M. (1995). Improved reasoning with Convince Me. *CHI '95 Proceedings: Short Papers*. Available: http://www.acm.org/sigchi/chi95/Electronic/documnts/shortppr/psk_bdy.htm

[13] Seel, N.M. (2001). Epistemology, situated cognition, and mental models: 'Like a bridge over troubled water'. *Instructional Science*, 29, 403-427.

[14] Tecuci, G. & Keeling, H. (1999). Developing an Intelligent Educational Agent with Disciple. *International Journal of Artificial Intelligence in Education*, 10, 221-237.

[15] Toth, J.A., Suthers, D. & Weiner, A. (1997). Providing Expert Advice in the Domain of Scientific Enquiry. In B. du Boulay & R. Mizoguchi (Eds.), *Artificial Intelligence in Education: Knowledge and Media in Learning Systems. Proceedings of AI-ED 97 World Conference on Artificial Intelligence in Education* (pp. 302-308). Amsterdam: IOS Press.

[16] van Gelder, T. (2002). Argument Mapping with Reason!Able. *American Philosophical Association Newsletter on Philosophy and Computers*, 85-90.

[17] Voss, J.F., Perkins, D.N. & Segal, J.W. (1991). Introduction. In J.F. Voss, D.N. Perkins & J.W. Segal (Eds.), *Informal Reasoning and Education*. Hillsdale, NJ: LEA.

[18] Voss, J.F. & Post, T.A. (1988). On the Solving of Ill-Structured Problems. In M.T.H. Chi, R. Glaser & M.J. Farr (Eds.), *The Nature of Expertise* (pp. 261-285). Hillsdale, NJ: Lawrence Erlbaum Associates.

[19] Wineburg, S. (2001). *Historical Thinking and Other Unnatural Acts: Charting the Future of Teaching the Past*. Philadelphia: Temple University Press.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Advanced Geometry Tutor: An intelligent tutor that teaches proof-writing with construction

Noboru Matsuda[*1] and Kurt VanLehn[*2]
mazda@cs.cmu.edu vanlehn@cs.pitt.edu

[*1]*Human-Computer Interaction Institute, Carnegie Mellon University*
[*2]*Learning Research and Development Center, University of Pittsburgh*

Abstract: Two problem solving strategies, forward chaining and backward chaining, were compared to see how they affect students' learning of geometry theorem proving with construction. In order to determine which strategy accelerates learning the most, an intelligent tutoring system, the Advanced Geometry Tutor, was developed that can teach either strategy while controlling all other instructional variable. 52 students were randomly assigned to one of the two strategies. Although computational modeling suggests an advantage for backwards chaining, especially on construction problems, the result shows that (1) the students who learned forward chaining showed better performance on proof-writing, especially on the proofs with construction, than those who learned backward chaining, (2) both forward and backward chaining conditions wrote wrong proofs equally frequently, and (3) the major reason for the difficulty in applying backward chaining appears to lie in the assertion of premises as unjustified propositions (i.e., subgoaling).

1 Introduction

Geometry theorem proving is one of the most challenging skills for students to learn in a middle school mathematics [1]. When a proof requires construction, the difficulty of the task increases drastically, perhaps because deciding which construction to make is an ill-structured problem. By "construction," we mean adding segments and points to a problem figure as a part of a proof. Our hypothesis is that teaching a general strategy for solving construction problems should help student acquire the skill, and that teaching a more computationally effective problem solving strategy might elicit faster learning.

For theorem proving that does not require construction, there are two common problem solving strategies: forward chaining and backward chaining. *Forward chaining* (FC for short) starts from given propositions and continuously applies postulates[1] forwards, that is, by matching the postulates' premises (antecedents) to proved propositions and instantiating its conclusions as newly proved propositions. This continues until FC generates a proposition that matches the goal to be proved. *Backward chaining* (BC for short) starts from a goal to be proved and applies postulates backwards, that is, by matching a conclusion of the postulate to the goal, then posting the premises that are not yet proved as new goals to be proved.

In earlier work [2], we found a semi-complete algorithm for construction that is a natural extension of backwards chaining, a common approach to proving theorems that do not involve construction. The basic idea is that a construction is done only if it is necessary for applying a postulate via backwards chaining. The same basic idea can be applied to the FC strategy.

We have conjectured that both BC and FC versions of the construction strategy are comprehensible enough for students to learn. A question then arises: would FC or BC better facilitate learning geometry theorem proving with construction? Furthermore, if there is any difference in the impact of different proof strategies, what would it be? This study addresses these questions.

[1] In this paper, a geometric "postulate" either means a definition, an axiom, or a proven theorem.

Earlier work suggests that there are pros and cons to both FC and BC as vehicles for learning proof-writing. From a cognitive-theories point of view, some claim that novice students would find it difficult to work with backward chaining [3, 4]. But others claim that novice to expert shift occurs from BC to FC [5, 6]. From a computational point of view, we found that FC is more efficient for theorem proving *without* construction, but BC is the better strategy for theorem proving *with* construction [2]. Yet we are lacking theoretical support to determine which one of these strategies better facilitates learning proof-writing with construction.

To answer the above questions, we have built two versions of an intelligent tutoring system for geometry theorem proving with construction, called the Advanced Geometry Tutor (AGT for short). The FC version teaches the construction technique embedded in forward chaining search. The BC tutor teaches the construction technique embedded in backward chaining search. We then assigned students to each tutoring condition, let them learn proof-writing under the assistance of AGT, and compared their performance on pre- and post-tests.

In the remaining sections, we first provide a detailed explanation of AGT. We then show the results from the evaluation study. We then discuss lessons learned with some implications for a future tutor design.

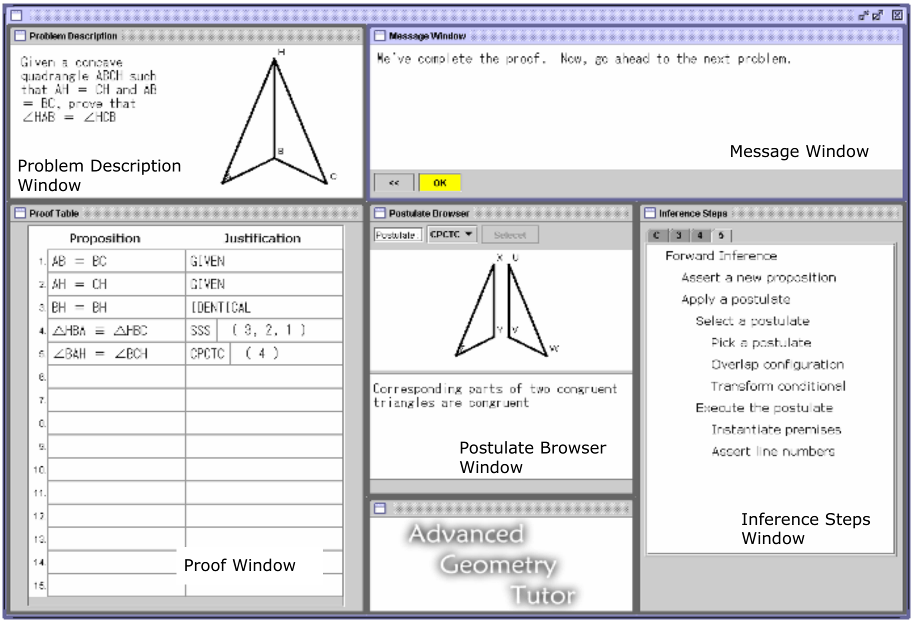
2 Advanced Geometry Tutor

This section describes the architecture of AGT. We first introduce the AGT learning environment. We then explain the scaffolding strategy implemented in AGT.

2.1 AGT learning environment

As shown in Figure 1, AGT has five windows each designed to provide a particular aid for learning proof writing.

Problem Description window: This window shows a problem statement and a problem figure. The problem figure is also used for construction. That is, the student can draw lines on the problem figure when it is time to do so.



Given a concave quadrangle ABCH such that AH = CH and AB = BC, prove that ∠HAB = ∠HCB

We've complete the proof. Now, go ahead to the next problem.

	Proposition	Justification
1.	AB = BC	GIVEN
2.	AH = CH	GIVEN
3.	BH = BH	IDENTICAL
4.	△HBA ≡ △HBC	SSS (3, 2, 1)
5.	∠BAH = ∠BCH	CPCTC (4)
6.		
7.		
8.		
9.		
10.		
11.		
12.		
13.		
14.		
15.		

Corresponding parts of two congruent triangles are congruent

Forward Inference
  Assert a new proposition
  Apply a postulate
    Select a postulate
      Pick a postulate
      Overlap configuration
      Transform conditional
    Execute the postulate
      Instantiate premises
      Assert line numbers

Figure 1: Advanced Geometry Tutor

Proof window: Although there are several ways to write a proof, we focus on a proof realized as a two-column table, a standard format taught in American schools, where each row consists of a proposition and its justification. A justification consists of the name of a postulate and, if the postulate has premises, a list of line numbers for the propositions that match its premises. The Proof window shown in Figure 1 shows a complete proof for the problem in the Problem Description window.

Message window: All messages from the tutor appear in this window. When the tutor provides modeling (explained in Section 2.2), the instructions that a student must follow appear here. When a student makes an error, feedback from the tutor also appears here. More importantly, this window is used for the students' turn in a tutoring dialogue, which sometimes consists of merely clicking the [OK] button. The dialogue history is stored, and the student is free to browse back and forth by clicking a backward [<<] and a forward [>>] button.

Postulate Browser window: The student can browse the postulates that are available for use in a proof. When the student selects a postulate listed in the browser's pull down menu, the configuration of the postulate, its premises, and its consequence are displayed. This window is also used by the tutor. As shown in Figure 1, when the tutor provides scaffolding on how to apply a particular postulate to a particular proposition, the configuration of the postulate changes its shape so that the student can see how the postulate's configuration should be overlapped with the problem figure.

Inference Step window: Although applying a postulate may seem like a single step to an expert, for a novice, it requires following a short procedure. The Inference Step window displays this procedure as a goal hierarchy of indented texts where each line corresponds to a single inference step in the postulate application procedure. The tutor highlights the inference step that is about to perform. The Inference Step window in Figure 1 shows inference steps performed to fill in the 5th row in the proof table.

2.2 Scaffolding strategy

The tutor uses both proactive and reactive scaffolding. Proactive scaffolding occurs before the step it addresses, whereas reactive scaffolding (feedback) occurs after the step.

To adapt the level of proactive scaffolding to the student, we apply Wood, Wood and Middleton's tutoring strategy [7], where the rule is, "If the child succeeds, when next intervening, offer less help; If the child fails, when next intervening, take over more control." The student's competence level for a step is maintained as follows. When the student correctly performs a step, the tutor increases the competence level. Conversely, when the student commits an error on a step, then the competence level for that step is decreased. Based on the student's competence level for a step, the tutor selects one of three types of proactive scaffolding: **Show-tell**: the tutor tells students what to do and actually performs the step. **Tell**: the tutor tells students what to do, but asks the student to perform the step. **Prompt**: the tutor only prompts the student to perform the step.

Reactive scaffolding (feedback) occurs immediately after a step. On the first failure to enter the step, the tutor provides minimal feedback (e.g., "Try again"). If the student fails again to enter this step, the tutor's help varies according to the student's competence level. For example, for an inference step for construction the tutor would say "Draw segments so that the postulate has a perfect match with the problem figure." When the student still fails to draw correct segments, the tutor lowers the competence level of that inference step and then provides a "Tell" dialogue, which generates a feedback message like "Draw new segments by connecting two points." If the students still can not make a correct construction, then the tutor provides more specific "Show-Tell" dialogue that would say "Connect points A and B." Note that this sequence roughly corresponds to a sequence of hints that starting from a general idea and becoming more concrete until very specific instruction (a bottom-out hint).

The tutor only gives hints when the student has made mistakes. Unlike many other tutors, AGT has no "Hint" button that students can press when they are stuck and would like a hint. However, the tutor does act like other tutors in keeping the student on a solution path. For instance, when there are several applicable postulates, the tutor will only let the student choose one that is part of a correct proof.

Although we chosen these instructional policies based on pilot testing and personal experience in tutoring geometry students, and we believe that they are appropriate for this task domain and these students, we have not compared them to other policies. Indeed, they were held constant during this study so that we could fairly evaluate the learning differences caused by varying the problem solving strategy that the tutor taught.

3 Evaluation

An evaluation study was conducted in the spring of 2004 to test the effectiveness of AGT and to examine an impact of different proof strategies on learning proof writing.

3.1 Subjects

52 students (24 male and 28 female) were recruited for monetary compensation from the University of Pittsburgh. The average age of the students was 23.3 (SD = 5.4). The students were randomly assigned to one of the tutor conditions where they used AGT individually.

3.2 Procedure and materials

Students studied a 9-page Geometry booklet, took a pre-test for 40 minutes, used an assigned version of AGT to solve 11 problems, and took a post-test for 40 minutes. Detailed explanations follow.

The booklet described basic concepts and skills of geometry theorem proving. It contained (1) a review of geometry proofs that explains the structure of geometry proofs and the way they are written, (2) a technique for making a construction, and (3) explanations of all 11 postulates used in the study. For each postulate, the booklet provided a general description of the postulate in English, a configuration of the postulate, a list of premises, and the consequence of the postulate. The booklet was available throughout the rest of the experiment, including all testing and training.

Pre- and post-tests consisted of three fill-in-the-blank questions and three proof-writing questions. The fill-in-the-blank questions displayed a proof-table with some justifications left blank and asked students to supplement those blanks. The proof-writing questions provided students with a proof table that was initialized with either a goal to be proven (for the FC condition) or given propositions (for the BC condition). There was one problem that did not require construction and two that required construction.

For both tutoring conditions, two tests, Test-A and Test-B, were used for the pre- and post-test. Their use was counterbalanced so that the half of the students took Test-A as a pre-test and Test-B as a post-test whereas the other half were assigned in a reversed order. Test-A and Test-B were designed to be isomorphic in the superficial feature of the questions and their solution structures, as well as the order of the questions in the test. Our intention was that working the tests would require applying exactly the same geometry knowledge in exactly the same order.

Besides the six problems used in the pre- and post-tests, 11 problems were used during the tutoring sessions. Among the 11 training problems, six required construction that could be done by connecting existing two points.

3.3 Results

A post evaluation analysis revealed that question 5 (a proof-writing problem) in Test-A and Test-B were not exactly isomorphic; question 5 in Test-B required additional application of

CPCTC (the Corresponding Part of Congruent Triangles are Congruent postulate) and SSS (the Side-Side-Side triangle congruent postulate). The students who took Test-B made more errors than those who took Test-A on question 5 hence there was a main effect of the test version on the pre-test: $t(50) = 2.32$; $p = 0.03$. When we excluded question 5 from both Test-A and Test-B, the main effect disappeared. Hence the following analyses exclude question 5 from both pre- and post-tests unless otherwise stated.

To evaluate an overall performance on the pre- and post-test, we used following variables to calculate individual students' post-test score. For fill-in-the-blank questions, the ratio of the number of correct answers to the number of blanks was calculated. For proof-writing questions, the ratio of correct proof statements to the length of a correct proof was calculated.

With these scores, students using the FC version of the tutor performed reliably better on the post-tests than students using the BC version. In an ANOVA, there was a main effect for the tutor on the post-test: $F(1,48) = 10.13$; $p<0.01$. The regression equation of the post-test score upon the pre-test score and the tutor condition was: Post-test = 0.52 * pre-test – 0.14 (if BC) + 0.50. Using the pre-test scores as a covariate in an ANCOVA, the adjusted post-test scores of 0.58 and 0.72 for the BC and FC students were reliably different. The effect size[2] was 0.72. In short, the FC students learned more than the BC students by a moderately large amount.

To see how the FC students outperformed the BC students, we conducted an item analysis by comparing scores on the fill-in-the-blank and proof-writing questions separately. For fill-in-the-blank questions, there were no significant differences between FC and BC students on the pre-test scores nor on post test scores. However, there was a main effect of the test (i.e., pre vs. post) on test scores for both FC and BC students: *paired*-$t(25) = 2.74$; $p = 0.01$ for FC, *paired*-$t(25) = 3.43$; $p < 0.01$ for BC. That is, both FC and BC students performed equally well on fill-in-blank questions, and they improved their performance equally well.

On proof-writing questions, the difference in pre-test was not significant ($t(50) = 0.91$; $p = 0.37$), but there was a main effect of tutor conditions for the post-test scores: $t(50) = 2.53$; $p = 0.02$. The effect size was 0.93.

The difference in the overall post-test scores between BC and FC students was thus mainly from the difference in proof-writing questions: the FC students wrote better proofs than BC students on the post-test. To understand how the FC students outperformed the BC students in proof writing, we further compared their performance on proof-writing with and without construction.

Since we excluded question 5, which was a construction problem, there was only one non-construction problem (question 4) and one construction problem (question 6). Figure 2 shows mean scores on these questions. The difference in the non-construction problem was not significant: $t(50) = 0.66$; $p = 0.51$, whereas the difference in the construction problem was significant: $t(50) = 2.89$; $p < 0.01$. That is, FC and BC students tied on non-construction problem, but FC students outperformed BC students on construction problem.

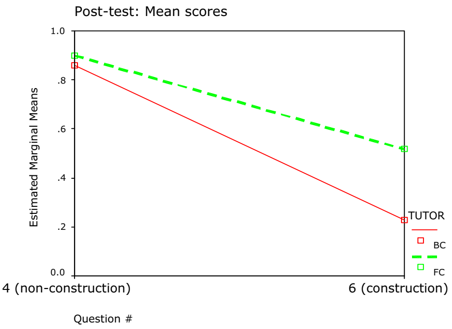


Figure 2: Mean scores on proof-writing for problem with and without construction

In order to narrow the locus of difference even further, we conducted 3 further analyses of the superiority of FC to BC. The analyses contrasted (1) the type of proof

[2] A ratio of the difference between FC and BC mean adjusted post-test scores to the standard deviation of the BC pre-test scores.

written for each problem, (2) the types of proof statements appeared in each proof, and (3) the quality of postulate applications used to compose each proof statement.

Before discussing these analyses, we need to introduce the scheme used to code proof statements. A proof statement, which is written on a single row in the proof table, consists of a proposition, a justification, and premises. A proof statement is said to be *on-path* when it is a part of a correct proof. An *off-path* proof statement is not a part of a correct proof, hence its proposition may or may not true, but the postulate used as a justification has a consequence that unifies with the proposition, and its antecedents unify with the premises listed in the justification. A *wrong* proof statement is neither on-path nor off-path.

Figure 4 shows the number of occurrence of each type of proofs. “OD” shows the number of proofs that were not written in the strategy taught (called TStrategy). The rest of this section excludes OD proofs. The figure clearly shows that FC students wrote more *correct* proofs, which by definition contain a tree of on-path proof statements connecting the givens to the top goal. FC and BC students were equally likely to write *wrong* proofs, which contains a tree of proof statements but the tree involves at least one proof statement that is not on-path. Aggregating *stuck* proofs where a proof does not contain a tree of proof statement, and *blank* proofs where no attempt for proof was made at all, BC students were more likely than FC students to fail in these ways.

Moving now to the statement-level analysis, there were 479 proof statements (215 and 264 in BC and FC conditions) appearing on the post-test. Of those, 400 were *reasonable* (i.e., either on-path or off-path) and 79 were *wrong* statements. 180 statements (92 in BC and 88 in FC) were *missing*, which means that they are necessary for a correct proof but were not mentioned at all. Figure 3 shows the frequency of each type of proof statements.

GRAMY often made off-path statements, especially when using FC to do constructions. However, the students seldom made off-path statements, especially in correct proofs, where only 3 off-path statements were written by FC students and no off-path statements were written by BC students. In incorrect proofs, off-path statements were slightly more frequent (19 for FC; 7 for BC), and FC students wrote more off-path proof statements than BC student ($\chi^2 = 8.52$; $df = 1$; $p < 0.01$). A further analysis revealed that all those off-path statements were made for postulate applications that did not involve construction. That is, when they made a construction, the students always write an on-path proof statement.

Another interesting phenomenon that can be read from Figure 3 is that BC students wrote wrong proof statements more frequently than FC students. Together with the fact mentioned earlier that the BC students tended to fail to start writing a proof (i.e., the Blank proofs in Figure 4), BC students apparently found it more difficult to write reasonable proof statements (on- and off-path) than FC students.

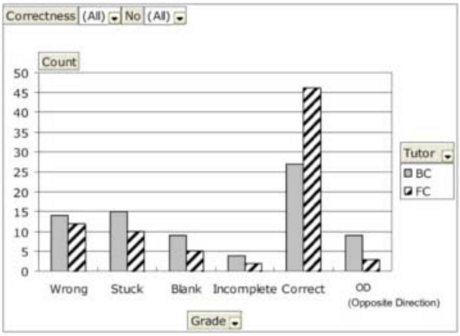


Figure 4: Classification of proofs

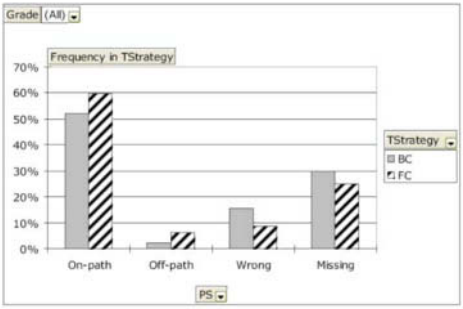
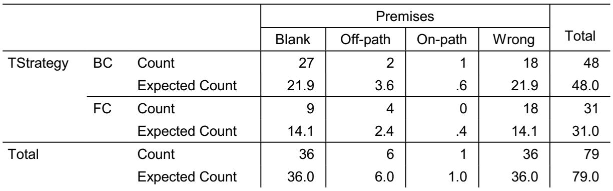


Figure 3: Classification of proof statements

As for the analysis on the quality of postulate applications, to investigate a reason for BC students having difficulty on writing proof statements, we coded each of the 79 wrong proof statements as a triple of independent codes of (1) the proposition, (2) the justification, and (3) the premises, which are three constituents of a proof statement. For each proof statement, we coded each instance of these constituents as on-path, off-path, wrong, or blank. We then ran 2 x 4 Contingency table analyses on each constituent to see if there was difference in the frequency of these constituents between BC and FC students.

For propositions and justifications, FC and BC did not display different frequency distributions. There was, however, a significant difference in the use of premises between FC and BC students. Figure 5 shows a 2 x 4 Contingency table on the use of premises. A Fisher's exact test on the table was 7.25 (p = 0.04), indicating a significant difference in the distribution of codes for premises. The BC students tended to leave the premises blank more often than the FC students. This tendency of leaving the premise blank was one reason for the inferiority of BC students in writing correct proofs compared to the FC students.



			Premises				Total
			Blank	Off-path	On-path	Wrong	
TStrategy	BC	Count	27	2	1	18	48
		Expected Count	21.9	3.6	.6	21.9	48.0
	FC	Count	9	4	0	18	31
		Expected Count	14.1	2.4	.4	14.1	31.0
Total		Count	36	6	1	36	79
		Expected Count	36.0	6.0	1.0	36.0	79.0

Figure 5: A 2 x 4 Contingency table on the use of premises

4 Discussion and Concluding Remarks

4.1 Learning proof-writing with construction

The first major contribution of this study is showing that proof-writing with construction can be taught with a technique that is a natural extension of theorem proving without construction. Although geometry construction is a difficult skill, perhaps even a creative one, it can be taught by conventional ITS technology, given that the tutor has an explicit problem solving strategy to teach that will solve construction problems.

Although there was not a main effect on accuracy of postulate applications measured with the fill-in-the-blank questions on the post test, some students (the FC ones) outperformed other (the BC ones) in proof-writing. This suggests that understanding domain principles (i.e., the concept of geometric postulates) is not sufficient for writing correct proofs. In addition, one must acquire proof-writing skills, and different kinds of instruction are differentially effective at facilitating this.

4.2 Impact of the different proof strategies on learning proof-writing

Despite the much higher computational demands of the FC version of the construction algorithm compared to the BC version, as documented in computational experiments with GRAMY [2], it turned out that FC students acquired more skill at construction than BC students. Our finding agreed with other empirical studies showing novice students' difficulty in applying backward chaining. It seems that problem solving complexity for a computer does not necessarily imply learning complexity for humans. Indeed, although both GRAMY and the students used both FC and BC, GRAMY always produces many off-path proof statements whereas the humans rarely did. This suggests that the humans are using knowledge or strategies not represented in GRAMY.

4.3 Difficulty in subgoaling

The BC students tended to get stuck at providing premises even when they picked a correct proposition and a postulate. It seems to be difficult for BC students to specify *subgoals* as the to-be-justified propositions that support a postulate application.

Subgoaling requires that the students write into the table one or more propositions (i.e., to satisfy the premises of a justification) that have yet to be proved. At the time they are entered into the proof table, those premises are not "true" assertions, but just hypothesis to be proved. This uncertainty may increase the chance of failure in backward chaining. Furthermore, those propositions are usually new in the proof table. Forward chaining, on the other hand, always enters propositions that are derived from known facts. Backward chaining differs mostly from forward chaining in this guess-and-try fashion in entering proof statements.

4.4 Implications for a future tutor design

A potential way to improve the BC tutor's efficacy is to intensify modeling and scaffolding on subgoaling for backward chaining. Although asserting unjustified propositions into a proof step was explicitly stated in the cognitive model of backward chaining utilized in AGT, the model was not effective in supporting the BC students in learning subgoaling.

The inadequacy of the BC tutor may also be due to a lack of instruction on *backtracking*. Backward chaining is essentially nondeterministic. For some goals, there are multiple equally plausible postulates whose consequences unify with the goal. Therefore, one must choose one of the postulates, try it, and if it does not work well, back-up to the choice point and choose another postulate. AGT acted as a more restricted tutor. Instead of allowing students to choose a postulate and possibly backup to this choice later, the tutor only allows them to choose an on-path postulate, so they never had to back up during training. This design principle is supported by an observation that the more the students flounder, the less opportunity they have for each cognitive skill to be exposed hence they achieve less learning [8]. For subgoaling, however, it might be necessary for students to understand that they are asserting hypotheses that could be wrong. Moreover, when applying backward chaining during the post-test, students may have had to choose among equally plausible postulates. This could cause confusion and consternation. Thus, it might be necessary to let students backtrack during training.

A related issue is to teach students to recover when they get stuck. Since the backward chaining strategy may lead them to an impasse, they should be taught what to do when they get stuck. AGT did not do this. Perhaps that is why the BC students often got stuck during the post-tests. AGT should train an ability to analyze the situation to identify an impasse, to diagnose the cause of the impasse, and to figure out an alternative way to avoid it by selecting a different path.

Acknowledgements: This research was supported by NSF Grants 9720359 and SBE-0354420.

Reference:

1. Senk, S.L., *How well do students write geometry proofs?* Mathematics Teacher, 1985. **78**(6): p. 448-456.
2. Matsuda, N. and K. VanLehn, *GRAMY: A Geometry Theorem Prover Capable of Construction.* Journal of Automated Reasoning, 2004. **32**(1): p. 3-33.
3. Trafton, J.G. and B.J. Reiser, *Providing natural representations to facilitate novices' understanding in a new domain: Forward and backward reasoning in programming.* Proceedings of the 13th Annual Conference of the Cognitive Science Society, 1991: p. 923-927.
4. Anderson, J.R., F.S. Bellezza, and C.F. Boyle, *The Geometry Tutor and Skill Acquisition*, in *Rules of the mind*, J.R. Anderson, Editor. 1993, Erlbaum: Hillsdale, NJ. p. 165-181.
5. Larkin, J., et al., *Expert and Novice Performance in Solving Physics Problems.* Science, 1980. **208**(4450): p. 1335-1342.
6. Chi, M.T.H., P.J. Feltovich, and R. Glaser, *Categorization and representation of physics problems by experts and novices.* Cognitive Science, 1981. **5**: p. 121-152.
7. Wood, D., H. Wood, and D. Middleton, *An experimental evaluation of four face-to-face teaching strategies.* International Journal of Behavioral Development, 1978. **1**(2): p. 131-147.
8. Anderson, J.R., et al., *Cognitive tutors: Lessons learned.* Journal of the Learning Sciences, 1995. **4**(2): p. 167-207.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Design of Erroneous Examples for ActiveMath

Erica Melis
German Research Institute for Artificial Intelligence (DFKI)
66123 Saarbruecken, Germany
melis@dfki.de

Abstract. The behaviorist view of learning that informs much of traditional schooling is not likely to invite students and teachers to see errors in a positive light. This is particularly true for mathematics. Our goal is to change this situation by including erroneous examples and other error-related learning opportunities in ActiveMath.

This paper investigates the systematic design of erroneous examples. For this, it analyzes the potential benefits that erroneous examples can have and distinguishes different presentation patterns. This analysis together with first experiences from school and from a university course with ActiveMath informs further research on effects, adaptive choice and presentation of erroneous examples in ActiveMath.

1. Introduction

The behaviorist view of learning that informs much of traditional schooling is not likely to invite students and teachers to see errors in a positive light. Behaviorism assumes that learning is enhanced when correct responses are rewarded (positive reinforcement) and incorrect ones are either punished or extinguished through lack of attention (withholding of positive reinforcement) [8]. Approaches to use errors as learning opportunities may help to overcome the traditional transmission view of mathematics teaching and learning.

Within the traditional framework, paying explicit attention to (mathematical) errors in class is even considered by many as dangerous since it could interfere with fixing the correct result in the student's mind. Indeed, the effectiveness of erroneous examples for different kinds of learners is an open issue and may depend on the individual learner. [15] investigated teachers' point of view on this and other issues with no conclusive results.

We know only of little research in psychology [12,5] which targets learning with erroneous examples. Some research in maths education addresses learning from errors that others made or that are deliberately introduced [1,9,14]. Mostly, these describe positive and creative reactions of teachers to student errors in the classroom which may be hard to implement in a learning environment. Hart [6] addresses the need to diagnose the learner's misconception (rather than the teacher's conceptions) for a proper reaction.

An intelligent system should use its potential to work with errors productively. One way to do this is through providing feedback on errors the student made. Another way is to include erroneous examples – a rather unusual type of exercises – into the learning experience.

This paper reports first steps and experiences with erroneous examples in the adaptive learning environment ACTIVEMATH [7]. This sets the stage for other computational issues such as generaltion of erroneous examples and adaptive choices. It investigates dimensions for the systematic design of erroneous examples. For illustration, the paper includes examples from our fraction course (school) and the derivatives course (university) which both are available online.

We would like to stress that the described design of erroneous examples does not primarily target the design of erroneous examples for lab experiments. Presumably, for this a more fine-grained tweaking is needed to obtain statistically significant results in a limited time-on-system.

2. Targeted Dimensions of the Learning Process

Including erroneous examples as exercises into a learning experience can serve several purposes:
(1) improvement of learner's motivation [14] and influence on students' attitudes towards failure and success.
(2) Proper understanding of concepts which includes conceptual change in case of a misconception [13] and understanding concept's boundaries. For concept learning, previous research indicates that people tend to use positive instances and ignore negative instances, see, e.g., [2]. This is an inefficient strategy. One measure to push students to look at negative instances is to require an explicit work on erroneous examples.
(3) Improve reasoning capabilities, e.g., the correct application of rules and the application of correct rules as well as hierarchical/structured problem solving.
(4) Train meta-reasoning including critical thinking, self-monitoring, and enforce self-explanation [12] to judge solution steps as correct or faulty. Meta-cognitive skills are required to overcome the barriers imposed by the student's prior knowledge and conceptions [10], and finding and correcting errors in an example can stimulate and prompt meta-cognitive activities. Critical thinking is sometimes neglected for mathematics and its applications. However, in real life people have to be able to judge whether a mathematical result is acceptable or to discover the conditions under which it is correct. They have to be able to find out the reason for an error. Learning should therefore, target this capability.
(5) Encourage exploration. Borasi [1] reports striking experiences on how even below-average students start questioning and exploring mathematics, when confronted with an error and encouraged to dwell on it.
(6) Change attitudes. In the traditional classroom there is not much room for being wrong, not even temporarily. Schoenfeld [11] reports that most students believe that if you can't solve a problem in a few minutes, you can't solve it at all. A mistake is interpreted as an ultimate failure and there is little room for experimentation (and debugging). When guessing, experimenting and playing

with partially correct conjectures are discouraged, the only remaining alternative for many students is getting stuck. Schoenfeld concludes that this attitude is an important factor in students' inability to cope with non-routine problems.

3. Design of Erroneous Examples

It is an art to design erroneous examples that include an obvious inconsistency and provoke conflicts. The most relevant variables for the design of an erroneous example are the actual error/misconception addressed and the example's actual presentation. The first is addressed implicitly in the examples below because it depends on the domain and on the typical errors that occur, the second is explicitly addressed.

There are several types of (typical) errors including buggy rules, misconceptions, and frequent slips such as wrong labels for quantities. A vast pedagogical literature about typical errors exists for school mathematics, e.g., for computation with fractions [4]. They collect and analyze procedural errors as well as misconceptions, e.g., [13].

As for the presentation, alternatives of the following Derivation Erroneous Example are described below.[1] In section 5 we summarize observations on when which presentation seems appropriate.

Eve wants to compute the derivative of the function: $y = \frac{1}{(1-2\cdot x)^2}$ for $x \neq \frac{1}{2}$. Her solution contains one or more errors. Please find the first error. [2]

Eve's solution: since $x \neq \frac{1}{2}$ holds, the function is differentiable in its domain. She uses the Chain Rule for computing the derivative.
The Chain Rule states that the derivative of a composite function $f \circ g$ can be calculated as follows $(f \circ g)'(x) = f'(g(x)) \cdot g'(x)$.
Eve chooses $f = \frac{1}{g^2}$ and $g = 1 - 2 \cdot x$

Now, Eve calculates the first factor $(\frac{1}{g^2})'$.

She begins with rewriting $f = \frac{1}{g^2}$ as $f = g^{-2}$ which leads to $f'(g) = (-2)g^3$

Then she calculates the second factor: $g'(x) = -2$.

Finally, she combines the factors as follows:
$(f(g(x))' = (f \circ g)'(x) = (-2) \cdot (1 - 2 \cdot x)^3 \cdot (-2) = 4 \cdot (1 - 2 \cdot x)^3$.

Erroneous Results vs Erroneous Worked Solution The Derivation Example shows an erroneous worked solution. An alternative presentation that can be generated consists of the erroneous result only.

Eve wants to compute the derivative of the function $y = \frac{1}{(1-2\cdot x)^2}$ for $x \neq \frac{1}{2}$. Her solution is $f'(x) = (1 - 2 \cdot x)^3$. Please find the error.

Correcting Errors vs Finding and Correcting In the first version, the errors are marked in the presentation of the erroneous example and the student is asked to correct them. In the second, the learner has to find the errors first and then correct. These alternatives can be produced automatically.

[1] This example is one from a set of erroneous examples we used in ActiveMath

[2] correcting the errors is requested subsequently

High-Level vs. Low-Level Questions Low-level questions ask for a particular step in the worked solution. For the Derivation Example, a multiple choice question (MCQ) with low-level choices asks to decide which of the following alternatives did actually occur in the erroneous example:

- the Chain Rule is not applicable here
- Eve differentiated $\frac{1}{g^2}$ wrongly
- Eve differentiated $1 - 2 \cdot x$ wrongly
- the computation of $(f \circ g)(x)$ is wrong
- a condition is missing.

A high-level question may cover several occurrences in a worked solution or ask for violated principles. An MCQ with high-level questions for the Derivation Example asks which type of error occures (first):

- a wrong derivation rule was chosen
- a rule was applied incorrectly
- an algebraic transformation was wrong
- the solution is correct only under certain conditions

MCQ vs Marking Both, MCQ and Marking exercises are *choice* exercises. Therefore, they can have the same representation from which either a low-level MCQ- or a Marking-interaction can be generated.

Describing as Erroneous vs Asking Student for Decision. The above Derivation Example indicates that Eve's solution is erroneous. Alternatively, the student is asked whether this solution is correct or not and why. If we decided for the second strategy, then it needs to include similar prompts for correct examples. A special case of 'Asking' addresses (missing) conditions (as for $x = \frac{1}{2}$ in the Derivation Example) and asks "in what circumstances could this result be considered correct?". Another special case of 'Asking' is the presentation of two solutions of the same problem for which one of them is flawed.

Feedback vs no Feedback In their study Grosse and Renkl [5] do not provide feedback to students. We think that feedback is crucial.

4. Adaptation wrt. Concept and Presentation

A user-adaptive system will choose erroneous examples (1) according to a meta-goal of learning, (2) according to a particular concept or rule the learner needs to understand and (3) appropriate wrt. difficulty. That is, the choice will depend on what the student model exhibits about the learner's misconceptions, buggy rules and attention, about his learning goals, and general capability.

For one and the same erroneous example there could be different reasons to choose it for different students. For instance, the Proof Example below can target the fringe conditions of division for one learner and target better attention and monitoring of his problem solving process for another student.

The learning goal and concepts can be served by the choice of an particular erroneous example and by the level (and content) of the questions/tasks for the learner. The difficulty is greatly influenced by the tasks and form.

5. Hypotheses and Observations for Erroneous Examples in Tests

This section summarizes first observations from two formative tests we have been running with erroneous examples in ActiveMath. The study with about 120 students at an under-privileged school (6th grade) did not allow for controlled conditions. For now, we can report observations only. Another study was performed in a seminar with 17 second to fourth year computer science students at the University of Saarland and we tested the acceptance and problems of working on erroneous proofs and erroneous derivation examples. In addition, a very mixed population (academics, non-academic adults and children) with 53 subjects was tested with erroneous proof of $2 = 1$ given below. The conditions were not controlled.

For the school test with ActiveMath, we interviewed teachers on the errors they would target for fractions. The resulting most frequent errors concern buggy addition procedure. These errors are addressed in erroneous examples of the current ActiveMath fraction course, for instance

Eve made a mistake when computing the sum of $\frac{1}{8}$ *and* $\frac{3}{8}$.
She computed $\frac{1}{8} + \frac{3}{8} = \frac{4}{16}$
Find her mistake! (and later: compute the sum of $\frac{1}{8}$ *and* $\frac{3}{8}$ *correctly).*

For the university test with ActiveMath, we employed the Derivation Erroneous Example and other examples with the following frequent errors for computing derivatives in terms of misconceptions and buggy rules

- wrong derivation rule used
- wrong application of a rule
- misconception of composite function, e.g., wrongly assumed commutativity
- misconception about variables or about dependency of variables
- misconception about fringe elements. No restriction of function domain
- wrong interpretation of the derivative in word problems

Moreover, we tested subjects with the erroneous Proof Example:

$$\begin{array}{lrcl}
\textit{Let} & a & = & b \\
\textit{multiply both sides of equation with } a & a^2 & = & ab \\
\textit{add } (a^2 - ab) \textit{ on both sides} & a^2 + a^2 - 2ab & = & ab - a^2 - 2ab \\
\textit{take out } (a^2 - ab) & 2(a^2 - ab) & = & 1(a^2 - ab) \\
\textit{division by } (a^2 - ab) \textit{ on both sides} & 2 & = & 1
\end{array}$$

To summarize, observations at *school* indicate that
(1) replacing examples by erroneous examples increased the motivation of almost all students
(2) students read/studied the erroneous examples more carefully than normal examples (which they obviously did not self-explain). That is, erroneous examples fought the problem that in maths classroom many students do not read instructions, definitions, examples carefully and do not spontaneously self-explain but immediately go to the exercises (performance-orientation)
(3) working with erroneous examples took longer than working with material that included examples instead. This indicates a conflict with the 'economy of learning' that prefers performance-oriented ways of learning.

Observations in the *university* experiment indicate that those students who were well-trained in logic and knowledgeable about epsilon-delta proofs, judged the task of finding and correcting some of the errors as 'too easy', even for a mistake for which other students struggle to discover it. Not surprisingly, this indicates that the choice of erroneous examples needs to be adapted to the learner's prerequisites and capabilities.

The test in which the above Erroneous Proof Example $2 = 1$ was used, was performed with a mixed population of 53 subjects 38 found the error and 15 did not. The Erroneous Proof gave rise to an unusually high attention (between 1 minute (for quick solvers only, the lowest dropout time was 5 minutes) and 20 minutes (one non-solver took even 45 minutes)!). 10 non-solvers rated erroneous examples exercises as a "useful way to learn mathematics". 5 non-solvers rated erroneous examples exercises as not useful. 28 solvers rated erroneous examples exercises as useful. 10 non-solvers rated erroneous examples exercises as not useful. A possible reason for this relative high attention and acceptance rate may be the obvious conflict $2 = 1$ which can be thought provoking.

For the different ways to present erroneous examples in §3 the following hypotheses were (partially) supported in the tests.

Erroneous Results vs Erroneous Worked Solution When given only an erroneous results, the task was more challenging. Students had to build possible solutions paths themselves. On the one hand, this seems to be more difficult than judging an erroneous worked example (and low-achieving students give up more easily). On the other hand, constructing alternative solution paths provides precious training. If a student is not able to find the error when given the result only, then presenting the erroneous worked solution can be the next choice.

We hypothesize that similar to the setting of self-explaining worked examples, the parts of the (erroneous) worked solution provides reminders and more support to a student than a full problem solving exercise.

Correcting Errors vs Finding and Correcting Finding and correcting errors was more difficult for (weak) students than only correcting errors with feedback. Finding and correcting involves two types of activities, the first one for reasoning and explaining and the second one for problem solving. That is, 'finding' required reasoning, self-explaining and/or careful watching each step in the example. This first interaction provides good learning opportunities. Therefore, only if a student cannot 'find' the error, she should obtain a correction-only presentation of the erroneous example.

High-Level vs. Low-Level Questions. Sometimes it is difficult to ask reasonable high-level questions other than 'is the result correct or incorrect?' To answer abstract questions, the student has to understand what the principles are and where they occur in the worked solution. Since high-level questions can be followed by lower-level questions or marking, the guidance itself is structured and thus, can support a more structured reasoning. This was observed in the university course. In the school test, this situation was observed too for tutor interventions but not yet tested with ACTIVEMATH.

Low-level MCQ vs Marking. We observed that marking seems to be more difficult for low-achieving students and can be somewhat more confusing at places (should a formula/result be marked or the reasoning/text that led to it?). We hypothesize that this is due to the smaller number and the explicit choice in case of MCQ.

Describing as Erroneous vs Asking Student for a Decision. Especially, the knowledgeable university students judged the more open format (in which they had to decide themselves about correctness) more interesting than a design stating that the solution is erroneous. We hypothesize that such a presentation will be more motivating for capable students. Moreover, the student has to be able to checking solutions and to inspect the problem solving space in order to be able to succeed with the problem.

Feedback As opposed to erroneous examples in the experiments of [5] ActiveMath provided orienting feedback for the finding phase as well as for the correction phase of erroneous examples. More detailed feedback is still under construction for the school course. For school students, the observations suggest that visualizations of the consequences of a learner's response may be needed in order to provoke cognitive conflicts.

6. Conclusion

Currently, erroneous examples are a rather unusual type of exercises in schools and in learning systems. However, they offer an interactivity that is primarily learning-oriented rather than performance-oriented.

We designed erroneous examples in ActiveMath with the long-term goal to improve the quality of learning at the cognitive and meta-cognitive level. This paper discussed several potential benefits of erroneous examples and different ways to design erroneous examples.

We reported the (informal) experiences from tests of ActiveMath with erroneous examples in a school and at the university.

Future Work

This work provides a basis for adapting to learning goals and students' capabilities. Future work will investigate in which situations erroneous examples are beneficial, how they have to be adapted for which learners, and how to generate useful feedback. Another problem is how to measure the learning effects that differ from performance improvement. This is important because performance is not the only dimension and may not even be the most important dimension of growth as discussed in section 2.

Acknowledgment

I thank Aiping Chen and Michael Dietrich for encoding erroneous examples on derivatives and on epsilon-delta proofs, Edgar Kessler for encoding the first erroneous examples for fractions and Martin Homik for his devoted activities in the university seminar.

This publication is partly a result of work in the context of the LeActiveMath project, funded under the 6th Framework Program of the European Community – (Contract IST-2003-507826). The author is solely responsible for its content.

References

[1] R. Borasi. Capitalizing on errors as "springboards for inquiry": A teaching experiment. *Journal for Research in Mathematics Education*, 25(2):166–208, 1994.

[2] L.E. Bourne, B.R. Ekstrand and B. Montgomery. Concept Learning as a Function of the Conceptual Rule and the Availability of Postive and Negative Instances. *Journal of Experimental Psychology*, 82(3): 538–544, 1969.

[3] A.T. Corbett and J.R. Anderson. Locus of feedback control in computer-based tutoring: Impact on learning rate, achievement and attitudes. In *Proceedings of CHI 2001*, pages 245–252. ACM, 2001.

[4] H.D. Gerster and U. Grevsmuehl. Diagnose individueller Schülerfehler beim Rechnen mit Brüchen. *Pädagogische Welt*, pages 654–660, 1983.

[5] C.S. Grosse and A. Renkl. Learning from worked examples: What happens if errors are included? In P. Gerjets, J. Elen, R. Joiner, and P. Kirschner (eds) *Instructional design for effective and enjoyable computer-supported learning*, pages 356–364. Knowledge Media Research Center, Tuebingen, 2004.

[6] K. Hart. Ratio and Proportion. In K. Hart (ed), *Children's Understanding of Mathematics*, pages 88–101, John Murray, London, 1981.

[7] E. Melis, E. Andrès, J. Büdenbender, A. Frischauf, G. Goguadze, P. Libbrecht, M. Pollet, and C. Ullrich. ACTIVEMATH: A generic and adaptive web-based learning environment. *International Journal of Artificial Intelligence in Education*, 1002(4):385–407, 2001.

[8] P.H. Miller. *Theories of Developmental Psychology.* Freeman, San Francisco, 1983.

[9] F. Oser and T. Hascher. Lernen aus Fehlern - Zur Psychologie des negativen Wissens. Schriftenreihe zum Projekt: Lernen Menschen aus Fehlern? Zur Entwicklung einer Fehlerkultur in der Schule, Pädagogisches Institut der Universität Freiburg, Schweiz, 1997.

[10] A.H. Schoenfeld. Explorations of students' mathematical beliefs and behavior. *Journal for Research in Mathematics Education*, 20(4):338–355, 1987.

[11] A.H. Schoenfeld. Explorations of students' mathematical beliefs and behavior. *Journal for Research in Mathematics Education*, 20(4):338–355, 1989.

[12] R.S. Siegler. Microgenetic studies of self-explanation. In N. Granott and J. Parziale (eds), *Microdevelopment, Transition Processes in Development and Learning*, pages 31–58. Cambridge University Press, 2002.

[13] S. Stafylidou and S. Vosniadou. The development of students' understanding of the numerical value of fractions. *Learning and Instruction*, 14:503–518, 2004.

[14] Ch. Strecker. Aus Fehlern lernen und verwandte Themen. http://www.blk.mat.uni-bayreuth.de/material/db/33/fehler.pdf, March 1999.

[15] P. Tsamir and D. Tirosh. In-Service Mathematics Teachers' Views or Errors in the Classroom. *International Symposium : Elementary Mathematics Teaching*, Prague, August, 2003.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

"Be bold and take a challenge": Could motivational strategies improve help-seeking?

Genaro REBOLLEDO MENDEZ, Benedict DU BOULAY and Rosemary LUCKIN
IDEAS Lab, Dept. of Informatics, University of Sussex
Brighton, BN1 9QH, UK
{G.Rebolledo-Mendez, B.Du-Boulay, R.H.Luckin}@sussex.ac.uk

Abstract. We are exploring whether the use of facilities aimed at improving the learner's motivation has an effect on learning food-chains and food-webs, but also on help-seeking behaviour. The M-Ecolab is a Vygotskyan intelligent learning environment that incorporates both cognitive and affective feedback by combining a cognitive model capable of providing written feedback at the cognitive and meta-cognitive level and a model-driven, considerate more-able partner who gives spoken, affective feedback. A preliminary study of the effects of the M-Ecolab in learning was carried out in a real-class situation. The results showed that learners in the M-Ecolab had significantly greater learning in their post-test scores than students in the control condition in which affective feedback was not available. Moreover, in the M-Ecolab, engaged students (those having an above-average use of the motivating facilities) tended to look more effectively both for quality and quantity of help resulting in more fruitful interactions.

1. Introduction

Education is a complex activity involving the complementary factors of the learner's cognitive and affective states. What is needed for the design of systems are models and theories that integrate the various cognitive and affective components [1]. Research in cognitive science has provided the means to understand the learning process better [2], and shown that meta-cognition is a crucial aspect of learning [3]. One of the meta-cognitive strategies that seems to have a great impact in learning is help-seeking [4]. This paper addresses the issue of the student's state of motivation and its interaction with help-seeking. In particular our project focuses on the effects of motivational scaffolding in the M-Ecolab, a Vygotskyan learning system for teaching children concepts related to food-chains and webs that in earlier versions has shown the effectiveness of scaffolding the learner's activities based on cognitive and meta-cognitive models [5]. The M-Ecolab provides a test-bed for modelling and reacting to different motivational states and allows an investigation of its effects on learning. The mechanisms for modelling the motivational state consist of estimations of the learners' effort, independence and confidence in the learning activities during the interaction with the system. The results of an exploratory study suggest that strategies aimed at improving the learners' motivation do indeed have an effect on learning and, interestingly, further data analysis suggest that the student's help-seeking behaviour might be improved with the use these strategies.

2. Help-seeking

Help-seeking allows learners to manage academic problems by keeping them actively involved in the learning situation [6]. The importance of this particular learning strategy lies in the fact that it can create means to acquire skills or knowledge not only for immediate but also for future application. Nelson-Le Gall [4] argues that help-seeking is a social activity and it is in social contexts that learners find the motivation to ask for help and contribute with their knowledge to assist others. To understand help-seeking, Nelson-Le Gall [4, 7] proposed a Vygotskyan framework consisting of the following steps:

1. Become aware of the need of help
2. Decide to seek help
3. Identify potential helper(s)
4. Use strategies to elucidate help
5. Evaluate help-seeking episodes

Research in intelligent tutoring systems has led to the development of different means to offer learners the help they need in their interactions. However, despite its benefits help-seeking is not always used effectively by students in learning environments [7]. To overcome this deficiency researchers have focused on providing means to create in learners an awareness of their need for help, as it is believed that successful students continually evaluate, plan and regulate their own academic progress. This self-awareness of the learning process, or meta-cognition, is a pre-requisite for help-seeking to occur but is not in itself obvious to some learners. Systems such as the Ecolab II [5], have tackled this issue by providing help at the meta-level, aimed at making the learners more aware of their help-seeking needs. Another more comprehensive approach has been the creation of a help-seeking behaviour model implemented as a set of production rules [8]. This model aims at developing meta-cognitive awareness by providing the learners, via a help-seeking agent, with feedback about their use of the help facilities.

Even if help-seeking awareness is the cornerstone of successful help-seeking behaviour, more research is needed in tutoring systems as it might not be the only process involved in successful help-seek behaviour. Nelson Le-Gall's model presented above, includes four more steps beyond simply awareness of the need for help that also have an important effect on help-seeking behaviour. The importance of the remaining processes, particularly step number two, is crucial for a successful help-seeking behaviour [9]. Of relevance for our work is the fact that Nelson Le-Gall's model presupposes a social context where not only the participation of the learner and a more-able partner are required, but also the learners' believes in their competence (ability) and actions (effort). It would be interesting to find out whether by creating an explicit, considerate more-able partner who is able to alter its spoken feedback based on an underpinning motivational model, more fruitful interactions occur. In particular we are interested in whether by scaffolding motivation, learners could not only learn and be made aware of their help-seeking deficiency but also advance their help-seeking behaviour by praising or encouraging their effort and independence. This is a novel approach as the effect that affective scaffolding could have in learning, and help-seeking behaviour in particular, has not yet been addressed.

We think that it is important to expand our knowledge in this area and we present preliminary results of the effects of motivational scaffolding in learning and its relationship with help-seeking in tutoring systems. We argue that within a simulated social context, provided by a computerised more-able partner informed about the particular cognitive and affective needs of the learner, students can progress from awareness to evaluation as in the model above.

3. Motivational scaffolding in the M-Ecolab

To shed some light onto this issue we have developed the M-Ecolab, an extension of earlier Ecolab software. Previous evaluations of the Ecolab system have illustrated the benefits of challenging the student and guiding, but not controlling, her intellectual extension [10] and of offering the learners help at the meta-level by making low-ability learners more aware of their help-seeking needs [11]. The success of this software is thought to derive from modelling the learner's cognitive and meta-cognitive traits. By analysing the learner's ability and collaborative support actions with the tutoring system, the Ecolab software is capable of altering the interactions offering different degrees of help and suggesting different learning activities (from a total of ten) to individual learners. The Ecolab provides cognitive help at four levels, the higher the level the greater the control taken by the system and the less scope there is for the pupil to fail [12].

Our approach for motivational scaffolding revolves around three motivational traits identified as key in learning contexts: effort, confidence and independence from the tutor [13]. The rationale of the M-Ecolab is that an underpinning model of the learner's motivation can be built by assessing her actions with the system and by considering the learner's cognitive and meta-cognitive state and relating them to motivational variables. The M-Ecolab also reacts accordingly by offering motivating elements that vary according to the perceived cause of de-motivation. Since the original Ecolab was based on a Vygotskyan model, a social environment was simulated by incorporating on-screen characters. The motivational model was implemented so that motivating scaffolding is available during the interaction with the software via a button within the interface, and is the rationale for the characters' behaviour.

The motivating facilities in the M-Ecolab consist of spoken feedback given by a more-able partner, a character called Paul. Since the system maintains a motivational model of the learner, Paul is able to alter his voice tone and gestures according to the perceived state of de-motivation in order to encourage the learner: be it to put more effort, to be more independent or to become more confident. There exist two classes of spoken feedback: pre- and post-activity. Pre-activity feedback informs the learner of the objectives of that learning activity whereas post-activity feedback offers motivating scaffolding making the learner reflect on her behaviour. The learner can listen to the spoken feedback as many times as she wants via a button on the interface. Additionally a quiz has been integrated as a motivating facility, but its activation depends on the learner and not on the underpinning motivational model. If activated, the quiz asks questions related to the food-chains topic. Wrong answers are not corrected but an indicator shows the number of correct and incorrect answers that the learner has tried-out during the interaction. Right answers are praised but a maximum of three correct answers is allowed per activity in order to avoid the learner to concentrate on the quiz more than the learning activity.

3.1 Ada and the M-Ecolab.

The following scenario illustrates a typical interaction with M-Ecolab:

Ada is a 10 year-old student who has not completed the 'Energy' activity in the M-Ecolab, but has attempted various eating actions without positive results. Ada then decides to choose a new activity. She clicks on the 'New Activity' button and a character appears introducing herself as Mrs. Johnson who tells Ada what the Ecolab is and what she is expected to do. To make things interesting, Mrs. Johnson prompts Ada to find what is inside a treasure-chest that can only be opened once she has collected the letters of a password.

Mrs. Johnson introduces Paul, who is a child from another school that has been successful in doing the Ecolab before. Paul then states the learning objectives for that particular activity (see Fig. 1). From now on a new button called ‘Treasure Chest’ appears on the interface. Ada clicks on the new button and discovers the empty password, the treasure chest and two buttons, one to call Paul and another to solve a quiz. She clicks on the Paul button causing Paul to repeat the learning objectives which direct her to the accompanying booklet. After having read the appropriate page of the booklet, Ada does correct and incorrect actions. Ada then notices a green tick appearing next to the ‘Activity button’ indicating that she has completed the activity. She decides to click on the ‘Activities’ button and Paul appears praising her efforts but stating that in the future she needs to ask for less help when she makes an error. Three models of her interaction are being created: a cognitive, a meta-cognitive and a motivational. According to the meta-cognitive model, the M-Ecolab suggests ‘Go on, learn about something new and the Ecolab will help you. Click on the activity that you want to do next:’. Ada selects a new activity called ‘Food 1’. A dialogue box appears with three choices of challenge and a suggestion ‘Be bold and take on a challenge’. Ada chooses challenge level 2 and then Paul, based on the motivational model states the objectives for that activity followed by a dialogue box indicating Ada to go to the booklet. She then continues working on the system, building more eating relationships, until she notices the green tick next to the ‘Activities’ button, indicating she has finished this activity. Once again she clicks on the ‘Activities’ button but this time Paul does not appear, as the motivational model believes she does not need more affective feedback. Ada continues with the activity called ‘Feeding 1’ creating more food-chains.

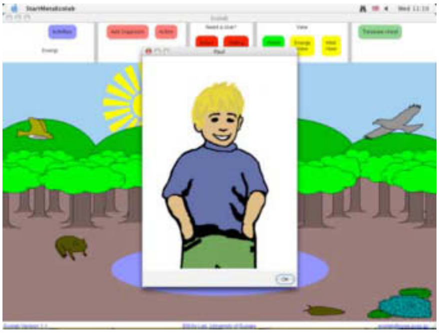


Fig. 1 An explicit more-able partner for the M-Ecolab

While Ada completes actions in the M-Ecolab, the system updates its three learner models: cognitive, meta-cognitive and motivational. These models consist of beliefs about how much she understands eating relationships, how much it believes she understands her own learning needs related to help-seeking, and also how much it believes she needs affective feedback. This information is used to adjust the affective post-activity feedback provided by the system according to the perceived degree of motivation, and to select the appropriate motivational trait that will be supported [13], prompting the character to alter his voice tone and gestures accordingly.

4. Preliminary evaluation of the M-Ecolab

To throw some light onto the issue of the influence of motivational scaffolding in the learner’s behaviour, an exploratory study of the effects of the M-Ecolab was conducted in a local primary school at the end of the academic year 2003-2004. We measured the students’ learning with the M-Ecolab using the same pre- and post-tests as in previous Ecolab evaluations [11]; the questions used in the learning tests were different from those of the quiz. The learners' motivation was assessed with an adaptation of Harter's test [14]. The participants were members of two fifth grade classes aged between 9-10. There were 10 students in the control condition, 5 girls and 5 boys and 19 learners in the experimental condition, 9 girls and 10 boys. All the students had been introduced to food-chains and food-webs prior to the study. The students were asked to complete a pre-test for 15 minutes and then a five-minute motivational questionnaire. Assistance was provided to the students who requested help to read the questions. Two weeks later, the M-Ecolab was demonstrated with the use of a video-clip showing its functionality. It was at this point that the researcher answered questions regarding the use of the software. One tablet PC was provided for each learner, with the appropriate version of the software (control = Ecolab, experimental = M-Ecolab). The students were then allowed to interact with it for 30 minutes. Immediately after the interactions, the pupils were asked to complete a post-test. Four weeks after the interaction the students were asked to complete a delayed post-test.

5. Results

This preliminary study looked at the effect that the two conditions had in increasing the student's learning in the Ecolab. To ensure that both conditions had a comparable level of knowledge about food-chains and food-webs, a t-test on the means of the pre-test was carried out showing a non-significant difference, see Fig. 2. In order to assess the overall learning gain in the M-Ecolab an analysis of covariance (ANCOVA) on the post- and delayed post-test data with three covariates: ability, motivation and performance on the pre-tests, indicates that the difference between the control and experimental groups is significant for both the post- and delayed post-test (post-test: F(4,28) = 9.013, p<.001; delayed post-test: F(4,27)=4.0,p<.02), see Fig. 2.

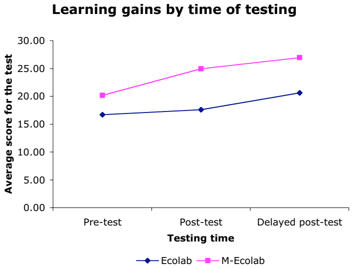


Fig. 2 Learning gains by time of testing

5.1 Motivation in the M-Ecolab

Motivation was assessed at two points during the study: the first time, during the pre-test using an adaptation of Harter's test [14] and the second, during the interaction using the underpinning motivational model embedded in the M-Ecolab [13]. Students having a below-average motivation according to Harter's tests were catalogued as less motivated. An analysis to contrast the learning gains in learners with low motivation between the two conditions was done with t-tests on the post-test's means. The results showed that learners with less motivation in the experimental condition yielded better learning than less motivated learners in the control group (t(13)= -2.280, p <. 05).

The underpinning motivational model in the M-Ecolab assesses motivation during the interaction using three motivational variables: effort, independence and confidence. A between-subjects analysis was carried out to assess the differences for the motivational variables within the two conditions. The results showed that there was not a significant difference in confidence or effort, but there was a significant difference in the independence values ($t(25) = 2.069$, $p < .05$) suggesting a greater independence for students in the control condition. These results were intriguing as it was expected that independent learners had gone beyond their intellectual capacities in the ZPD; however, judging from the findings in the post-test scores, it was clear that independence did not yield better learning outcomes in the Ecolab condition.

5.2 Help-seeking in the M-Ecolab

In order to deepen the analysis of independence, an examination of the type of help that learners had during their interactions was undertaken. In the M-Ecolab, less-independent students had greater degrees of help and showed lower effort in individual activities during the interaction with the system. Less independent students were more likely to be found in the experimental group (n=11) than in the control group (n=2). However, despite being less independent, students in the experimental group were more successful in their pre-, post-test learning gain as revealed by a within-subjects test ($t(18) = -3.815$, $p < .01$). To throw more light on the aspect of help that accounted for these learning gains, an analysis of help-seeking was undertaken distinguishing quantity from quality of help and trying to understand the nature of collaborative support requested by the students:

- Participants having an above-average quantity of help, whether provided by the software or requested by the student, were catalogued as having "lots" of help, otherwise as having "little" help.
- The mean level of help was calculated for all the participants, if learners received an average level of help greater than the group's mean, more quality of help, they were considered to have "deep" help or "shallow" otherwise.

Results indicated that students in the M-Ecolab condition who had little help (less quantity) increased their learning from the pre- to the post-test ($t(9)=-3.381,p<.01$). Moreover, participants requesting for deep help (more quality) in the M-Ecolab condition accounted for better performance in a within-subjects design ($t(8)=-4.239,p<.01$) than those in the control group. These results suggest that in M-Ecolab quality rather than quantity of help accounts for a greater impact in learning. A further between-subjects analysis of the differences in the means in the post-test scores for students using deep help (see Table 1) shows that there is a significant different between the two conditions ($t(16)=-2.5443$, $p<.05$), suggesting better learning experiences in the M-Ecolab. Table 1 also shows that the mean values for quality (with values ranging from one to four) and quantity of help (measured through the number of clicks on the help button) were greater for the M-Ecolab condition, albeit not significantly, suggesting that the effect of motivating facilities prompted the learners to request for more quality and quantity of help.

	Quality of help (mean level of help requested)	Quantity of help (clicks on help button)	Mean post-test scores
M-Ecolab (n=10)	3.12	17.50	24.70
Ecolab (n=7)	3.03	1186	18.86

Table 1. Mean values for students requesting deep help.

In order to have an insight into the role of the motivating facilities provided by the M-Ecolab, participants having an above-average request for motivating facilities were catalogued as "engaged". The results of a paired-samples test indicated that engaged students in the M-Ecolab accounted for a greater learning from the pre- to the post-test (t(8) = -4.807, p < .01), but not the disengaged students. Although there is not a significant difference in learning when comparing the post-test between the engaged and disengaged groups, the evidence suggest that better learning outcomes occurred when more quality of help was selected, replicating previous findings [5]. However, the evidence also suggests that there is a tendency in the experimental group, particularly among engaged students, to look for a greater quality of help, although this result is not significant (t(16) = -1.934, p = .071).

6. Discussion and conclusions

This exploratory study has presented evidence that motivating facilities might improve help-seeking behaviour in the M-Ecolab. The results suggest that learners using the M-Ecolab had more learning gains in both, post- and delayed post-tests than those in the control group. The M-Ecolab is a Vygotskyan system whose aim is to develop the learners' ZPD [15] implying, among other things, a more independent behaviour on the part of the learners. An analysis of the motivational variables that make-up the underpinning motivational model suggested that the motivational variable with greater differences across conditions was independence, being the students in the M-Ecolab less independent and at the same time more successful in their post-test scores. This finding was intriguing as it was expected that a more independent behaviour could lead to greater learning gain. As in M-Ecolab independence is modelled in terms the cognitive model's belief about the learners' need of help, the lack of independence (the need of more help) prompts the system to provide motivating feedback aimed at creating awareness about help-seeking.

A further analysis of the help-seeking behaviour showed that, in correspondence with previous evaluations, it was the learners who asked for greater quality of help rather than just more help those who achieved better learning outcomes. The evidence suggests that within the experimental condition, learners making more use of the motivating facilities were also those requesting higher quality of help. The findings of earlier Ecolab evaluations [11] highlighted the importance of providing the learner with challenging activities but also of offering help at the meta-level, so making the learners more aware of their help-seeking needs, which is consistent with the process of teaching within the ZPD [15]. This is also valid in the M-Ecolab but now it also seems that by having an explicit more-able partner learners, particularly those seeking the more-able partner's assistance, seemed to engage in more fruitful interactions. It also seems that the factor prompting the learners to ask for the help they need is the presence of the motivating facilities, as it was engaged students who improved their learning most.

It is recognised that there were two main problems with this pilot study. The first was the small number of participants; the second was the limited amount of time that was allowed for the interaction. With longer interaction time a richer analysis could be made of the effects of the more-able partner in the learning process, ruling-out the possibility of the 'novelty effect' that the motivating facilities could create in short interactions. If motivation goes beyond the novelty effect, longer interactions could improve an incipient collaborative setting between the learner and Paul. If Paul, who is already able to change his tone of voice, could also able to alter his facial expressions the feedback provided by him could create more productive interactions. With longer interactions times it could be possible to elucidate whether the pupils do pay more attention to Paul and ultimately decided to follow his advice. Future evaluation will overcome these shortcomings and also reveal whether an adaptive model that does not present motivating facilities if they are not necessary will work as well for all ability pupils. Work also needs to be done to find a relationship between meta-cognition and the various traits that affect motivation, particularly confidence, as Tobias and Everson [16] suggest that it is likely that high displays of meta-cognition reduce anxiety, hence increasing confidence. There are more possibilities open beyond the current investigation, such as making Paul say the feedback at the meta-level. By doing so it might be possible to investigate whether the learner advances through more steps in Nelson-Legall's model [4]. It would be interesting to define and further explore, how increases in help-seeking capability in the learner improves learning.

Acknowledgments

We would like to thank Mr. M. Ayling and Greenway Primary School in Horsham, UK. This work has been partially granted by the Conacyt and Veracruzana University.

References

[1] Pintrich, P.R. and T. Garcia, *Student goal orientation and self-regulation in the college classroom.* Advances in Motivation and Achievement, 1991. **7**: p. 371-402.

[2] Lajoie, S.P. and S.J. Derry, *Computers as cognitive tools.* 1993, Hillsdale, NJ: Lawrence Erlbaum Associates.

[3] Flavell, J.H., *Metacognition and cognitive monitoring: A new area of cognitive-developmental inquiry.* American Psychologist, 1979. **34**: p. 906-911.

[4] Nelson-Le Gall, S., *Help-seeking: An understudied problem solving skill in children.* Developmental Review, 1981. **1**: p. 224-246.

[5] Hammerton, L. and R. Luckin. *How to help? Investigating children's opinions on help.* in *10th International Conference on Artificial Intelligence in Education.* 2001. San Antonio, TX: St Mary's University, p. 22-33.

[6] Nelson-Le Gall, S. and L. Resnich, *Help seeking, achievement motivation and the social practice of intelligence in school*, in *Strategic help-seeking. Implications for learning and teaching*, S.A. Karabenick, Editor. 1998, Lawrence Erlbaum Associates, Inc.: London.

[7] Aleven, V., et al., *Help Seeking and Help Design in Interactive Learning Environments.* Review of Educational Research, 2001. **73**(2): p. 277-320.

[8] Aleven, V., et al. *Toward tutoring help seeking.* in *Seventh International Conference on Intelligent Tutoring Systems, ITS 2004.* 2004. Brasil: Spring Verlag, p. 237-239.

[9] Ryan, A.M. and P.R. Pintrich, *Achievement and social motivational influences on help seeking in the classroom.*, in *Strategic help seeking. Implications for learning and teaching*, K. S.A., Editor. 1998, Earlbaum: Mahwah. p. 117-139.

[10] Luckin, R., *´Ecolab´: Explorations in the Zone of Proximal Development, PhD Thesis*, in *School of Cognitive & Computer Sciences, University of Sussex.* 1998, University of Sussex: Brighton, BN1 9QH.

[11] Luckin, R. and L. Hammerton, *Getting to know me: helping learners understand their own learning needs through Metacognitive scaffolding*, in *Proceedings of the sixth Conference on Intelligent Tutoring Systems.* 2002, Berlin : Springer: Biarritz, France.

[12] Wood, D.J. and H.A. Wood, *An experimental evaluation of four face to face teaching strategies.* International Journal of Behavioural Development, 1978. **1**: p. 131-147.

[13] Rebolledo, G. *Motivational Modelling in a Vygotskyan ITS.* in *Artificial Intelligence in Education.* 2003. Sydney, Australia: IOS Press, p. 537-538.

[14] Harter, S., *A new self report scale of intrinsic versus extrinsic orientation in the classroom: motivational and informational components.* Developmental Psychology, 1981. **17**(3): p. 300-312.

[15] Vygotsky, L.S., *Mind in society : The development of higher psychological processes.* 1978, Cambridge, MA: Harvard University Press.

[16] Tobias, S. and H.T. Everson, *Knowing what you know and what you don´t: further research on metacognitive knowledge monitoring.* 2002, College Entrance Examination Board: New York. p. 25.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Educational Data Mining: a Case Study

Agathe MERCERON[*] and Kalina YACEF[+]
[*]*ESILV - Pôle Universitaire Léonard de Vinci, France*
[+]*School of Information Technologies - University of Sydney, Australia,*
Agathe.Merceron@devinci.fr, kalina@it.usyd.edu.au

Abstract. In this paper, we show how using data mining algorithms can help discovering pedagogically relevant knowledge contained in databases obtained from Web-based educational systems. These findings can be used both to help teachers with managing their class, understand their students' learning and reflect on their teaching and to support learner reflection and provide proactive feedback to learners.

1 Introduction

Web-based educational systems collect large amounts of student data, from web logs to much more semantically rich data contained in student models. Whilst a large focus of AIED research is to provide adaptation to a learner using the data stored in his/her student model, we explore ways to mining data in a more collective way: just as a human teacher can adapt to an individual student, the same teacher can also learn more about how students learn, reflect and improve his/her practice by studying a group of students.

The field of Data Mining is concerned with finding new patterns in large amounts of data. Widely used in Business, it has scarce applications to Education. Of course, Data Mining can be applied to the business of education, for example to find out which alumni are likely to make larger donations. Here we are interested in mining student models in a pedagogical perspective. The goal of our project is to define how to make data possible to mine, to identify which data mining techniques are useful and understand how to discover and present patterns that are pedagogically interesting both for learners and teachers.

The process of tracking and mining such student data in order to enhance teaching and learning is relatively recent but there are already a number of studies trying to do so and researchers are starting to merge their ideas [1]. The usefulness of mining such data is promising but still needs to be proven and stereotypical analysis to be streamlined. Some researchers already try and set up some guidelines for ensuring that ITS data can be usefully minable [2] out of their experience of mining data in the project LISTEN [3].

Some directions start to emerge. Simple statistics, queries or visualisation algorithms are useful to give to teachers/tutors an overall view of how a class is doing. For example, the authors in [4] use pedagogical scenarios to control interactive learning objects. Records are used to build charts that show exactly where each student is in the learning sequence, thus offering to the tutor distant monitoring. Similarly in [5], students' answers to exercises are recorded. Simple queries allow to show charts to teachers/tutors of all students with the exercises they have attempted, they have successfully solved, making tutors aware of how students progress through the course. More sophisticated information visualisation techniques are used in [6] to externalise student data and generate pictorial representations for course instructors to explore. Using features extracted from log data and marks obtained in the final exam, some researchers use classification techniques to predict student performance fairly accurately [7]. These allow tutors to identify students at risk and provide advice ahead of the final exam. When student mistakes are recorded, association rules algorithms can be used to find mistakes often associated together [8]. Combined with a genetic algorithm, concepts mastered together can be identified using student scores[9].

The teacher may use these findings to reflect on his/her teaching and re-design the course material.

The purpose of this paper is to synthesize and share our various experiences of using Data Mining for Education, especially to support reflection on teaching and learning, and to contribute to the emergence of stereotypical directions. Section 2 briefly presents various algorithms that we used, section 3 describes our data, section 4 describes some patterns found and section 5 illustrates how this data is used to help teachers and learners. Then we conclude the paper.

2 Algorithms and Tools

Data mining encompasses different algorithms that are diverse in their methods and aims [10]. It also comprises data exploration and visualisation to present results in a convenient way to users. We present here some algorithms and tools that we have used. A data element will be called an individual. It is characterised by a set of variables. In our context, most of the time an individual is a learner and variables can be exercises attempted by the learner, marks obtained, scores, mistakes made, time spent, number of successfully completed exercises and so on. New variables may be calculated and used in algorithms, such as the average number of mistakes made per attempted exercise.

Tools: We used a range of tools. Initially we worked with Excel and Access to perform simple SQL queries and visualisation. Then we used Clementine[11] for clustering and our own data mining platform for teachers, Tada-Ed [12], for clustering, classification and association rule (Clementine is very versatile and powerful but Tada-Ed has pre-processing facilities and visualisation of results more tailored to our needs). We used SODAS [13] to perform symbolic data analysis.

Data exploration and visualisation: Raw data and algorithm results can be visualised through tables and graphics such as graphs and histograms as well as through more specific techniques such as symbolic data analysis (which consists in creating groups by gathering individuals along one attribute as we will see in section 4.1). The aim is to display data along certain attributes and make extreme points, trends and clusters obvious to human eye.

Clustering algorithms aim at finding homogeneous groups in data. We used k-means clustering and its combination with hierarchic clustering [10]. Both methods rest on a distance concept between individuals. We used Euclidian distance.

Classification is used to predict values for some variable. For example, given all the work done by a student, one may want to predict whether the student will perform well in the final exam. We used C4.5 decision tree from TADA-Ed which relies on the concept of entropy. The tree can be represented by a set of rules such as: *if* $x=v_1$ *and* $y> v_2$ *then* $t= v_3$. Thus, depending on the values an individual takes for, say the variables x and y, one can predict its value for t. The tree is built taking a representative population and is used to predict values for new individuals.

Association rules find relations between items. Rules have the following form: *X -> Y, support* 40%, *confidence* 66%, which could mean '*if students get X incorrectly, then they get also Y incorrectly*', with a support of 40% and a confidence of 66%. Support is the frequency in the population of individuals that contains both *X* and *Y*. Confidence is the percentage of the instances that contains *Y* amongst those which contain *X*. We implemented a variant of the standard Apriori algorithm [14] in TADA-Ed that takes temporality into account. Taking temporality into account produces a rule *X->Y* only if exercise *X* occurred before *Y*.

3 A case study: Logic-ITA student data

We have performed a number of queries on datasets collected by the Logic-ITA to assist teaching and learning. The Logic-ITA is a web-based tutoring tool used at Sydney

University since 2001, in a course taught by the second author. Its purpose is to help students practice logic formal proofs and to inform the teacher of the class progress [15].

3.1 Context of use

Over the four years, around 860 students attended the course and used the tool, in which an exercise consists of a set of formulas (called premises) and another formula (called the conclusion). The aim is to prove that the conclusion can validly be derived from the premises. For this, the student has to construct new formulas, step by step, using logic rules and formulas previously established in the proof, until the conclusion is derived. There is no unique solution and any valid path is acceptable. Steps are checked on the fly and, if incorrect, an error message and possibly a tip are displayed. Students used the tool at their own discretion. A consequence is that there is neither a fixed number nor a fixed set of exercises done by all students.

3.2 Data stored

The tool's teacher module collates all the student models into a database that the teacher can query and mine. Two often queried tables of the database are the tables *mistake* and *correct_step*. The most common variables are shown in ***Table 1***.

Table 1. Common variables in tables mistake and correct_ step

login	the student's login id
qid	the question id
mistake	the mistake made
rule	the logic rule involved/used

line	the line number in the proof
startdate	date exercise was started
finishdate	date exercise was finished (or 0 if unfinished)

4 Data Mining performed

Each year of data is stored in a separate database. In order to perform any clustering, classification or association rule query, the first action to take is to prepare the data for mining. In particular, we need to specify two aspects: (1) what element we want to cluster or classify: students, exercises, mistakes? (2) Which attributes and distance do we want to retain to compare these elements? An example could be to cluster students, using the number of mistakes they made and the number of correct steps they entered. Tada-ed provides a pre-processing facility which allows to make the data minable. For instance, the database contains lists of mistakes. If we want to group that information so that we have one vector per student, we need to choose how the mistakes should be aggregated. For instance we may want to consider the total number of mistakes, or the total number of mistakes per type of mistake, or a flag for each type of mistake, and so on.

4.1 Data exploration

Simple SQL queries and histograms can really allow the teacher get a first overview of the class[8, 15]: what were the most common mistakes, the logic rules causing the most problems? What was the average number of exercises per student? Are there any student not finishing any exercise? The list goes on.

To understand better how students use the tool, how they practice and how they come to master both the tool and logical proofs, we also analysed data, focussing on the number of attempted exercises per student. In SODAS, the population is partitioned into sets called symbolic objects. Our symbolic objects were defined by the number of attempted exercises and were characterized by the values taken for these newly calculated variables: the number of successfully completed exercises, the average number of correct steps per attempted exercise, the average number of mistakes per attempted exercise. We obtained a number of tables to compare all these objects. An example is given in Table 2, which compares objects according to the number of successfully completed exercises.

Table 2. Distribution of students according to the number of attempted exercises (row) and the number of completed exercises (column) for year 2002.

Finish/Attempt	0	1	2	3	4	5	6	7	8	9	10	11	12	14	15	16	19	20	21	26
1	46	54																		
2	13	23	65																	
3	6	11	39	44																
4-6	4	8	27	19	29	10	2													
7-10	3		6	18	36	12	18	3	3											
11-15			16	16	16	21	5	5			11		5	5						
16 +			17												17		17	33		17

For example, the second line says that, among the students who have attempted 2 exercises, 13% could not complete any of them, 23% could complete one and 65% could complete both. And similarly for the other lines.

Using all the tables, we could confirm that the more students practice, the more successful they become at doing formal proofs[16]. Interestingly though, there seems to be a number of exercises attempted bove which a large proportion of students finish most exercises. For 2002, as little as two attempted exercises seem to put them on the safe side since 65% of the students who attempted 2 exercises were able to finish them both.

4.2 Association rules

We used association rules to find mistakes often occurring together while solving exercises. The purpose of looking for these associations is for the teacher to ponder and, may be, to review the course material or emphasize subtleties while explaining concepts to students. Thus, it makes sense to have a support that is not too low. The strongest rules for 2004 are shown in Table 3. The first association rule says that if students make mistake *Rule can be applied, but deduction incorrect* while solving an exercise, then they also made the mistake *Wrong number of line references given* while solving the same exercise. Findings were quite similar across the years (2001 to 2003).

Table 3. Association rules for Year 2004.

M11 ==> M12 [sup: 77%, conf: 89%] M12 ==> M11 [sup: 77%, conf: 87%] M11 ==> M10 [sup: 74%, conf: 86%] M10 ==> M12 [sup: 78%, conf: 93%] M12 ==> M10 [sup: 78%, conf: 89%] M10 ==> M12 [sup: 74%, conf: 88%]	M10: Premise set incorrect M11: Rule can be applied, but deduction incorrect M12: Wrong number of line reference given

4.3 Clustering and visualisation

We applied clustering to try and characterize students with difficulties. We looked in particular at those who attempted an exercise without completing it successfully. To do so, we performed clustering using this subpopulation, both using (i) k-means in TADA-Ed, and (ii) a combination of k-means and hierarchical clustering of Clementine. Because there is neither a fixed number nor a fixed set of exercises to compare students, determining a distance between individuals was not obvious. We calculated and used a new variable: the total number of mistakes made per student in an exercise. As a result, students with similar frequency of mistakes were put in the same group. Histograms showing the different clusters revealed interesting patterns. Consider the histogram shown in Figure 1 obtained with TADA-Ed. There are three clusters: 0 (red, on the left), 1 (green, in the middle) and 4 (purple, on the right). From other windows (not shown) we know that students in cluster 0 made many mistakes per exercise not finished, students in cluster 1 made few mistakes and students in cluster 4 made an intermediate number of mistakes. Students making many mistakes use also many different logic rules while solving exercises, this is shown with the vertical, almost solid lines. On the other hand, another histogram (Figure 2) which displays exercises against students, tells us that students from group 0 or 4 have not attempted more exercises than students from group 1, who make few mistakes. This suggests that these students try out the logic rules from the pop-up menu of the tool one after the other while solving exercises, till they find one that works.

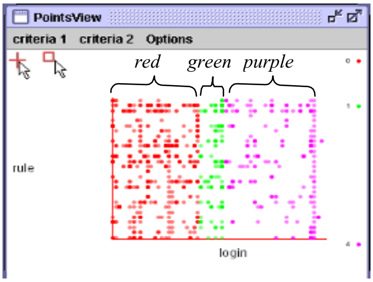


Figure 1. Histogram showing, for each cluster of students, the rules incorrectly used per student

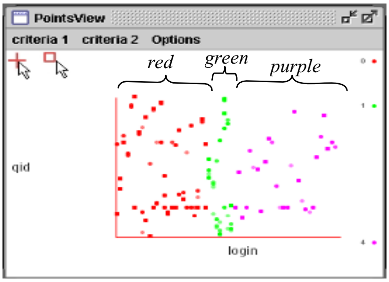


Figure 2. Histogram showing, for each cluster of students, the exercise attempted per student

Note: Since the article is printed in black and white, we superimposed information about where the colors are located.

4.4 Classification

We built decision trees to try and predict exam marks (for the question related to formal proofs). The Decision Tree algorithm produces a tree-like representation of the model it produces. From the tree it is then easy to generate rules in the form `IF condition THEN outcome`. Using as a training set the previous year of student data (mistakes, number of exercises, difficulty of the exercises, number of concepts used in one exercise, level reached) as well as the final mark obtained in the logic question), we can build and use a decision tree that predicts the exam mark according to the attributes so that they can be used the following year to predict the mark that a student is likely to obtain.

Table 4. Some results of decision tree processing. Accuracy of mark prediction using simple rounding of the mark (on 84 students).

Attributes and type of pre-processing	**Accuracy of mark**	**Accuracy of pass/fail**	**Diff. Avg (sd) real/predicted**	**Rel. error**
Number of distinct rules in each exercise* Number of exercises per performance type^	51.9%	73.4%	-0.2 (1.7)	11%
Number of distinct rules* Sum of lines entered correctly in each exercise	46.8%	87.3%	-0.5 (1.9)	18%
Number of exercises per nb of rules (interval)* Different performance achieved ^	45.6%	86.1%	-0.4 (1.8)	14%
Number of different length of exercises# Different performance achieved ^	43%	88.6%	0.14 (1.5)	8%
Number of exercises per performance type^ Sum of lines entered correctly in each exercise	44.3%	86.1%	-0.3 (1.7)	13%
Number of exercises per performance type^ Sum of rules used correctly (incl. repetition)	44%	86.1%	0.1 (1.9)	10%
Sum of rules used correctly (incl. repetition)	43%	87.3%	-0.22 (1.8)	13%
Sum of lines entered correctly in each exercise	43%	87.3%	-0.22 (1.8)	13%
Mistakes, in any form of pre-processing	<20%			

* in order to avoid overfitting we have grouped number of rules into intervals: [0-5], [6-10], [10+] .
# for the same reason, the number of steps in exercises was grouped into intervals of 5.
^ Performance types were grouped into 3 types: unfinished, finished with mistakes, finished without mistake.

There are a very large number of possible trees, depending on which attributes we choose to do the prediction and how we use them (ie the type of pre-processing we use). We investigated this on different combinations, using 2003 year as training data (140 students) and 2004 year as test data (84 students). After exam results, the 2004 population did very slightly better than the 2003 one, but not with a statistical difference. For each combination we calculated accuracy at different granularity. Table 4 shows some of the results we obtained: the second column shows the percentage of mark accuracy (a prediction is deemed accurate when the rounded value predicted coincides with the real mark). The third column shows the percentage of accuracy of pass/fail predictions. The fourth column shows the average difference between the predicted exam value and the real exam value, and the standard deviations (which are the same as the root mean squared prediction error). The last column shows the relative squared error. Marks ranged from 0 to 6.

The most successful predictors seemed to be the number of rules used in an exercise, the number of steps in exercises and whether or not the student finished the exercises. Interestingly, these attributes seemed to be more determining than the mistakes made by the student, regardless of how we pre-process them.

5 Supporting teachers and learners

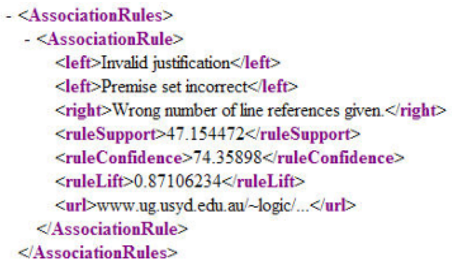
5.1 Pedagogical information extracted

The information extracted greatly assisted us as teachers to better understand the cohort of learners. Whilst SQL queries and various histograms were used during the course of the teaching semester to focus the following lecture on problem areas, the more complex mining was left for reflection between semesters.

- Symbolic data analysis revealed that if students attempt at least two exercises, they are more likely to do more (probably overcoming the initial barrier of use) and complete their exercises. In subsequent years we required students to do at least 2 exercises as part of their assessment (a very modest fraction of it).
- Mistakes that were associated together indicated to us that the very concept of formal proofs (ie the structure of each element of the proof, as opposed to the use of rules for instance) was a problem. In 2003, that portion of the course was redesigned to take this problem into account and the role of each part of the proof was emphasized. After the end of the semester, mining for mistakes associations was conducted again. Surprisingly, results did not change much (a slight decrease in support and confidence levels in 2003 followed by a slight increase in 2004). However, marks in the final exam continued increasing. This leads us to think that making mistakes, especially while using a training tool, is simply part of the learning process and was supported by the fact that the number of completed exercises per student increased in 2003 and 2004.
- The level of prediction seems to be much better when the prediction is based on exercises (number, length, variety of rules) rather than on mistakes made. This also supports the idea that mistakes are part of the learning process, especially in a practice tool where mistakes are not penalised.
- Using data exploration and results from decision tree, one can infer that if students do successfully 2 to 3 exercises for the topic, then they seem to have grasped the concept of formal proof and are likely to perform well in the exam question related to that topic. This finding is coherent with correlations calculated between marks in the final exam and activity with the Logic Tutor and with the general, human perception of tutors in this course. Therefore, a sensible warning system could look as follows. Report to the lecturer in charge students who have completed successfully less than 3 exercises. For those students, display the histogram of rules used. Be proactive towards these students, distinguishing those who use out the pop-up menu for logic rules from the others.

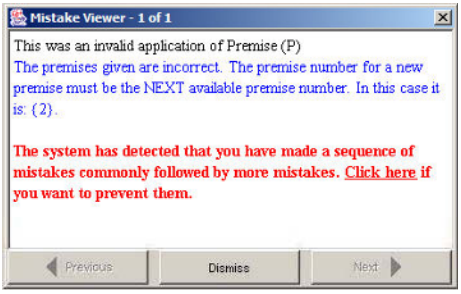
5.2 ITS with proactive feedback

Data mining findings can also be used to improve the tutoring system. We implemented a function in Tada-Ed allowing the teacher to extract patterns with a view to integrate them in the ITS from which the data was recorded. Presently this functionality is available for Association Rule module. That is, the teacher can extract any association rule. Rules are then saved in an XML file and fed into the pedagogical module of the ITS. Along with the pattern, the teacher can specify an URL that will be added to the feedback window and where the teacher can design his/her own proactive feedback for that particular sequence of mistakes. The content of the page is up to the teacher. For instance for the pattern of mistakes A, B -> C, the teacher may want to provide explanations about mistakes A and B (which the current student has made) and review underlying concepts of mistake C (which the student has not yet made).



```
- <AssociationRules>
  - <AssociationRule>
      <left>Invalid justification</left>
      <left>Premise set incorrect</left>
      <right>Wrong number of line references given.</right>
      <ruleSupport>47.154472</ruleSupport>
      <ruleConfidence>74.35898</ruleConfidence>
      <ruleLift>0.87106234</ruleLift>
      <url>www.ug.usyd.edu.au/~logic/...</url>
    </AssociationRule>
  </AssociationRules>
```

Figure 3. XML encoded patterns



This was an invalid application of Premise (P)
The premises given are incorrect. The premise number for a new premise must be the NEXT available premise number. In this case it is: {2}.

The system has detected that you have made a sequence of mistakes commonly followed by more mistakes. Click here if you want to prevent them.

Figure 4. Screen shot of mistake viewer

The structure of the XML file is fairly simple and is shown in ***Figure 3***. For instance, using our logic data, we extracted the rule saying that if a student makes the mistakes "Invalid justification" followed by "Premise set incorrect" then s/he is likely to make the mistake "Wrong number of references lines given" in a later step (presently there is no restriction on the time window). This rule has a support of 47% and a confidence of 74%. The teacher, when saving the pattern, also entered an URL to be prompted to the student.

The pedagogical module of the Logic Tutor then reads the file and adds the rule to its knowledge base. Then, when the student makes these two initial mistakes, s/he will receive, in addition to the relevant feedback on that mistake, an additional message in the same window (in a different color) advising him/her to consult the web page created by the teacher for this particular sequence of mistakes. This is illustrated by ***Figure 4***.

This allows the tutoring system to send proactive messages to learners in order to try and prevent mistakes likely to occur later, based on patterns observed with real students.

5.3 Support for student reflection

Extracting information from a group of learners is also extremely relevant to the learner themselves. The fact that learner reflection promotes learning is widely acknowledged [17]. The issue is how to support it well. A very useful way to reflect on one's learning is to look up what has been learned and what has not yet been learned according to a set of learning goals, as well as the difficulties currently encountered. We are seeking here to help learners to compare their achievements and problems in regards to some important patterns found in the class data. For instance, using a decision tree to predict marks, the student can predict his/her performance according to his/her achievements so far and have the time to rectify if needed. Here more work needs to be done to assess how useful this prediction is for the student.

6 Conclusion

In this paper, we have shown how the discovery of different patterns through different data mining algorithms and visualization techniques suggests to us a simple pedagogical policy. Data exploration focused on the number of attempted exercises combined with classification led us to identify students at risk, those who have not trained enough. Clustering and cluster visualisation led us to identify a particular behaviour among failing students, when students try out the logic rules of the pop-up menu of the tool. As in [7], a timely and appropriate warning to students at risk could help preventing failing in the final exam. Therefore it seems to us that data mining has a lot of potential for education, and can bring a lot of benefits in the form of sensible, easy to implement pedagogical policies as above.

The way we have performed clustering may seem rough, as only few variables, namely the number and type of mistakes, the number of exercises have been used to cluster students in homogeneous groups. This is due to our particular data. All exercises are about formal proofs. Even if they differ in their difficulty, they do not fundamentally differ in the concepts students have to grasp. We have discovered a behaviour rather than particular abilities. In a different context, clustering students to find homogeneous groups regarding skills should take into account answers to a particular set of exercises. Currently, we are doing research work along these lines.

References

[1] Beck, J., ed. *Proceedings of ITS2004 workshop on Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes*. Maceio, Brazil (2004).

[2] Mostow, J. "Some Useful Design Tactics for Mining ITS Data" in *Proceedings of ITS2004 workshop Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes*, Maceio, Brazil (2004).

[3] Heiner, C., J. Beck, & J. Mostow. "Lessons on Using ITS Data to Answer Educational Research Questions" in *Proceedings of ITS2004 workshop Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes*, Maceio, Brazil (2004).

[4] Gueraud, V. & J.-M. Cagnat. "Suivi à distance de classe virtuelle active" in *Proceedings of Technologies de l'Information et de la Connaissance dans l'Enseignement Supérieur et l'Industrie (TICE 2004)*, pp 377-383, UTC Compiègne, France (2004).

[5] Duval, P., A. Merceron, M. Scholl, & L. Wargon. "Empowering learning Objects: an experiment with the Ganesha Platform" in *Proceedings of ED-MEDIA 2005*, Montreal, Canada (2005).

[6] Mazza, R. & V. Dimitrova. "CourseVis: Externalising Student Information to Facilitate Instructors in Distance Learning" in *Proceedings of 11th International Conference on Artificial Intelligence in Education (AIED03)*, F. Verdejo and U. Hoppe (Eds), Sydney: IOS Press (2003).

[7] Minaei-Bidgoli, B., D.A. Kashy, G. Kortemeyer, & W.F. Punch. "Predicting student performance: an application of data mining methods with the educational web-based system LON-CAPA" in *Proceedings of ASEE/IEEE Frontiers in Education Conference*, Boulder, CO: IEEE (2003).

[8] Merceron, A. & K. Yacef. "A Web-based Tutoring Tool with Mining Facilities to Improve Learning and Teaching" in *Proceedings of 11th International Conference on Artificial Intelligence in Education.*, F. Verdejo and U. Hoppe (Eds), pp 201-208, Sydney: IOS Press (2003).

[9] Romero, C., S. Ventura, C. de Castro, W. Hall, & M.H. Ng. "Using Genetic Algorithms for Data Mining in Web-based Educational Hypermedia Systems" in *Proceedings of AH2002 workshop Adaptive Systems for Web-based Education*, Malaga, Spain (2002).

[10] Han, J. & M. Kamber, *Data mining: concepts and techniques*, San Francisco: Morgan Kaufman (2001).

[11] SPSS, *Clementine*, www.spss.com/clementine/ (accessed 2005)

[12] Benchaffai, M., G. Debord, A. Merceron, & K. Yacef. "TADA-Ed, a tool to visualize and mine students' online work" in *Proceedings of International Conference on Computers in Education, (ICCE04)*, B. Collis (Eds), pp 1891-1897, Melbourne, Australia: RMIT (2004).

[13] SODAS, http://www.ceremade.dauphine.fr/~touati/sodas-pagegarde.htm (accessed 2003)

[14] Agrawal, R. & R. Srikant. "Fast Algorithms for Mining Association Rules" in *Proceedings of VLDB*, Santiago, Chile (1994).

[15] Yacef, K., "The Logic-ITA in the classroom: a medium scale experiment". *International Journal on Artificial Intelligence in Education*. 15: p. 41-60 (2005).

[16] Merceron, A. & K. Yacef, "Mining Student Data Captured from a Web-Based Tutoring Tool: Initial Exploration and Results". *Journal of Interactive Learning Research (JILR)*. 15(4): p. 319-346 (2004).

[17] Boud, D., R. Keogh, & D. Walker, eds. *Reflection: Turning Experience into Learning*. Kogan Page: London (1985).

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Adapting Process-Oriented Learning Design to Group Characteristics

Yongwu Miao and Ulrich Hoppe
Institute of Computer Science and Interactive Systems
University of Duisburg-Essen, Germany

Abstract. IMS LD supports the design of personalized learning by adapting activities and other process elements to personal characteristics. This paper presents a new idea to adjust activities and elements to group characteristics for supporting 'groupalized' or group-adapted learning. This may help to improve effectiveness and efficiency of group-based collaborative learning. Our approach to formalize adaptive learning designs for groupalization is based on an extension of IMS LD. As a first "proof of concept", a fully implemented prototype system is presented.

Introduction

Adaptation to a learner's personal learning objectives, interests, preferences, performances, and other characteristics is a key challenge in many research areas concerning learning technologies such as Intelligent Tutoring Systems [12] and Adaptive Educational Hypermedia [1][11]. Typical approaches to personalized learning are adjusting contents, their structures, and presentations to personal characteristics. At present, there is a trend in learning technologies that the emphases shift from content to activities. The publication of IMS Learning Design [3], an international standard designed to promote exchange and interoperability of content with a focus on facilitating reuse of instructional strategies, can be considered as a positive step forward in this direction. IMS LD provides a "meta-language" which can be used to describe a wide range of pedagogical approaches. A pedagogical approach can be described in IMS LD as a set of structured learning activities in a formalized process model.

One of the objectives of IMS LD is to support personalization [5]. Based on IMS LD several attempts have been made to support personalized learning by adapting activities and other process elements within a unit of learning to personal characteristics/requirements [2][8][9][10]. The basic idea is that a process model can be reused by multiple learners. Learning processes will be adapted to different personal characteristics and profiles of learners. IMS LD provides mechanisms for formalizing adaptive learning process models, which can be automatically executed at run-time system.

In this paper, we propose a new idea -- to support group-based collaborative learning by adapting activities and elements to the characteristics of groups. Corresponding to the term of personalized learning, we use the term 'groupalized learning'. Groupalized or group-adapted learning is a kind of learning design tailored for individual groups according to the diversity in group characteristics. This leads to a number of new research questions: whether it is generic and significant to develop flexible collaborative learning process models to suit for different groups; how such an adaptive learning process model can be formalized; what a group model should be developed for this purpose; what factors should be taken into account for adaptation; what elements within an activity such as tools and content can be adjusted and how this can be done; what support is needed in run-time systems; how automatic adaptation and human-involved adaptation can be integrated; what is the relation with personalized learning and so on. Our assumption is that, like personalized learning processes for individuals, groupalized learning processes may help to improve the learning of groups, if adaptive learning designs can be appropriately specified.

In order to go in this direction we have to take the first step. The focus of this paper is on presenting the generic approach and supporting the specification of adaptive learning design for groupalized learning. In the next section, we use a scenario to analyze the characteristics of group-based learning processes and show an example of adaptive learning design. Then, we identify the requirements to specify adaptive learning designs. After presenting our approach and a prototype system to support formalizing adaptive learning designs for groupalization, we conclude our work with indicating future directions.

1. Characteristics of Adaptive Learning Design

In order to help us to analyze the characteristics of adaptive, group-based collaborative learning processes, we will introduce and discuss a scenario that is based on an open issue given by a teacher. The design rationale of this learning design is to create conflicts among students and engage them into interactions to resolve the conflicts and to avoid the situations in which some participants dominate the discussion and others just behave as listeners.

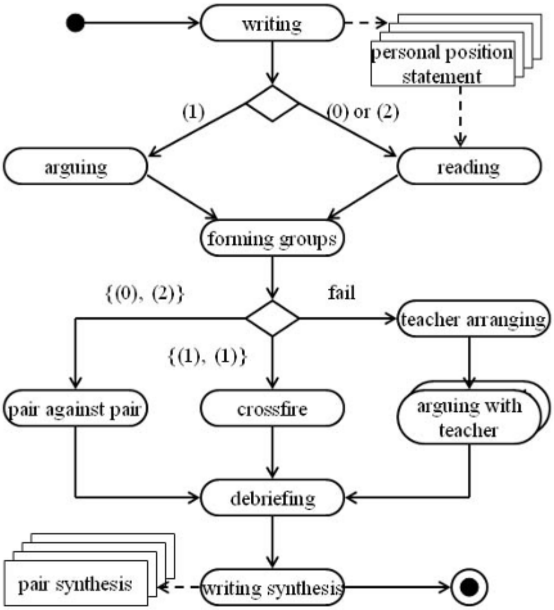
1.1 A Scenario

In a class a teacher gives an open issue to students and requires students using a “pair argue” method. Students are experienced in applying this method because they usually conduct discussion by adopting this method. Each student has a stable partner in the class. Toni and Darina are two students as a pair. First each student writes a position statement independently. When both students in a pair finish writing, they will check whether they have the same position. If having opposite positions, they will argue and try to resolve conflicts. Otherwise, they will exchange position statements with another pair in which both students have a common agreement as well but an opposite position to theirs. Toni and Darina have opposite opinions and after arguing no one can persuade the other. Then, each pair looks for another pair to conduct a discussion according to the following rules: either two homogeneous pairs (both students have the same position) with different positions, or two heterogeneous pairs. If some pairs cannot find an appropriate combination (e.g., all pairs take the same position), the teacher will arrange specific activities for these pairs, for example, assigning some pairs to take the role of objectors and facilitating a debate with an opposite role. After forming a big group, Toni and Darina will continue their debate with an assistant. Finally, the teacher facilitates a debriefing discussion in the class. After the class, each pair has to write a synthesis as homework.

1.2 An Example Model

Figure 1 shows a UML activity diagram that specifies the pedagogical approach, the “pair argue” method described in the scenario. This process model presents an e-learning version of an adaptive learning design with ten activities, two branches, and two sets of artifacts. Among activities, “writing” is an individual activity; “forming groups” is a supportive activity done by an automatic agent; “teacher arranging” is a supportive activity also performed by the teacher; “debriefing” is a session of all students and the teacher. The rest of activities are pair activities. In this diagram, we use the notations (0) and (2) to represent two types of homogeneous pairs, respectively. The notation (1) represents a heterogeneous pair. The notations {(0), (2)} and {(1), (1)} represent two kinds combinations: one is the combination of two homogeneous pairs with different positions and the other is a combination of any two heterogeneous pairs. The notation "fail" refers the event that the automatic agent cannot make appropriate combinations for some pairs. This event results in the intervention of a teacher. It is note that some details are ignored for focusing on the control flow and data flow of the model. For example, certain tools may be used in activities such as chat tool, shared whiteboard, shared text editor, issue-based argumentation tool, A/V conferencing tool and so on.

Figure 1: An Example Model of Adaptive Learning Designs for Groupalization



1.3 Characteristics of Adaptive Learning Design

From this simple example, we can see some characteristics of adaptive learning design for groupalization.

First, a pedagogical method can be described as a process model that can be repeatedly executed by multiple groups at the same time or at different time. There may be synchronization points in this process. However, it is possible for different groups to take the same process model with different paths at a different pace. The adaptation components are primarily learning activities rather than content. In fact, content is defined within activities. However, in this example there is no content defined in the process model.

Secondly, although there are individual activities and community activities, most activities are group activities. A group as a whole goes through the process from the beginning to the end. Each group activity will be done collaboratively and will terminate when the whole group rather than an individual finishes the tasks. Furthermore, each group will have static and dynamic characteristics while executing the model. In this example, according to the positions of both students in a pair, each pair must fall into one of three categories: (0), (1), and (2). This can be regarded as a kind of dynamic characteristic of the group.

Thirdly, certain characteristics will be used to determine the learning path of each group. In this example, there are two checkpoints and alternate paths in the diagram are based on whether the positions of two students in a pair are the same or not. In addition, multiple alternative activities are available at each checkpoint and each group with certain characteristics will take appropriate activities. In the example, the first branch specifies two options: one for homogeneous pairs to select the "reading" activity and the other for heterogeneous pairs to select the "arguing" activity. In the second branch, the category of pairs is used as a primary factor to determine the path of a pair as well, although it is not the unique factor in this case. Sometimes users may also adapt activities to groups' characteristics.

2. Requirements for Formalizing Adaptive Learning Design

As mentioned before, the emphasis of this paper is on formally representing adaptive learning design for groupalization. According to the characteristics of adaptive learning designs, a formal process modeling language should have mechanisms to represent the following aspects to support adaptation for groupalization.

2.1 Representing Pedagogical Models

A process modeling language should have mechanisms to represent a whole learning process, not only including content but also including roles, learning activities, services, control flow, data flow, etc.. Such a description should be a computer-executable model. The components and their relationships within the model can be used to decide upon adaptation.

2.2 Representing Group Models

A group model used for adaptation should not only maintain generic information about the group (e.g., name, members, creation time, form-policy, etc), but also maintain dynamic information (e.g., active activities, finished activities, intermediate outcomes, etc). Such information captures the characteristics of groups that is used for adaptation.

2.3 Representing Adaptation Models

The process modeling language should provide mechanisms to define the adaptation logic as well as adaptation actions. The former is responsible for relating information available models (e.g., process model, activity model, content model, group model, etc) and assessing whether adaptations are required. The latter refers to specifying the very actions (e.g., showing/hiding activities, forming high-level groups, making configuration of tools, setting property value, making content visible/invisible, etc) that need to be effected by the system for a given adaptation to be achieved. In addition, it must allow the designer, when desired, to pass the control over the adaptation process to users.

3. Supporting Adaptive Learning Design for Groupalized Learning

IMS LD [3] is a meta-language for modeling learning designs. When trying to use IMS LD to model group-based collaborative learning processes, we see several difficulties. In order to solve the problems, we developed a computer supported collaborative learning (CSCL) scripting language by extending IMS LD. The generic considerations and a whole picture about the CSCL scripting language have been described in [7]. This paper focuses on discussing an additional issue in detail -- how the extended language can be used to support the formalization of adaptive learning designs for groupalization. Our generic approach is to reuse the mechanisms provided by IMS LD originally for constructing adaptive rules for personalization. Concretely speaking, IMS LD level B and level C introduce mechanisms of properties, conditions and notifications, which can be used to specify arbitrarily complex dynamic behaviors of a system [6]. This section presents how these mechanisms are reused and extended to meets the requirements identified in the last section.

3.1 Reusing IMS LD to Specify Pedagogical Models

Rather than attempting to capture the specifics of many pedagogical models, IMS LD does this by providing a generic and flexible language that is designed to enable express many different pedagogical models. It can be used to express the pedagogical meaning and functionality of the different data elements within the context of a learning design. By using IMS LD, a learning design can be represented in the following way. People with certain roles work individually or collaboratively towards certain outcomes by performing a set of structured activities within associated environments, in which appropriate learning objects and services are available. In addition, IMS LD provides mechanisms to formalize activity models, content models, user models, role model, etc. These models are useful for specifying adaptation. Therefore, we primarily use IMS LD to specify pedagogical models.

3.2 Introducing Groups

The conceptual framework of IMS LD does not include the concept of group. Within the framework, role is an entity relevant to the group. The notation of role can be used to model group in many learning designs. However, mixing up these two different concepts may lead to serious mistakes when modeling some collaborative learning processes. For example, in the scenario if a role is defined for each student pair, how many roles have to be defined in the example learning design model? In fact, in this learning design model we can define a role "student pair". Then each pair as a whole takes this role. In order to enable the simple and intuitive modeling of group-based collaborative learning, the concept of group is explicitly introduced into the conceptual framework. A group can have individual members and sub-groups. All groups with their person members are structured as a directed-acycle-diagram. A group as a whole can take a role. A group has attributes such as identifier, name, max-size, min-size, person members, super-groups, sub-groups, engaged roles, form-policy, disband-policy, dynamic/static, creation-time and so on. Furthermore, a group model not only encapsulates general information about the group, but also maintains a "live" account of the group's actions within the system.

In order to support modeling dynamic characteristics or pedagogy-specific characteristics, we add concepts of local group property and global group property in the framework of our process modeling language. Like a local person property, a local group property has a different value for every group in a run. The property is owned by the run of the unit-of-learning, specifying a value per group. In the example model the combination of the pair opinions can be modeled by using a local group property. Like a global person property, a global group property can have a different value for each group, independent of the different executed instances of units of learning. The group entity owns the property specifying the portfolio of the group. For example, the "pair synthesis" produced in the last activity of the example model can be modeled as a global group property, because the value of this property (a synthesis) may need to be stored permanently by the run-time system as a kind of group information. Such information will be used by other learning designs.

Properties can be used to define property-groups as well. Therefore, a group model can be specified with static characteristics and dynamic characteristics.

3.3 Enriching Adaptation Logics and Adaptation Actions

IMS LD allows for describing personalization aspects within a learning design, so that the activities and content within a unit-of-learning can be adapted based on the preferences, portfolio, prior knowledge, educational needs and situational circumstances of users. At level B, an adaptive rule can be represented as a condition clause in the following way:

if <condition> then <actions> else <actions>

In order to express adaptation logics and adaptation actions, IMS LD provides limited operations on process elements, called element operations in this paper. There are two categories of element operations. The purpose of the first category of element operations is to get the state of a given element at run-time (like the method get() in JAVA) such as datetime-activity-started, users-in-role, when-property-value-is-set, and activity-completed. If a parameter such as an activity, a role, or a property is past to the element operations described above, the element operations will return a value such as a time, a set of user identifiers, or a Boolean. The element operations in the second category will effect on the state of a given element at run-time (like the method set() in JAVA) such as set-property, change-property-value, and show/hide (changing the status of the "isvisible" attribute of the given element). In addition, IMS LD provides {expression} schema group to facilitate specifying complex adaptation logics for personalization.

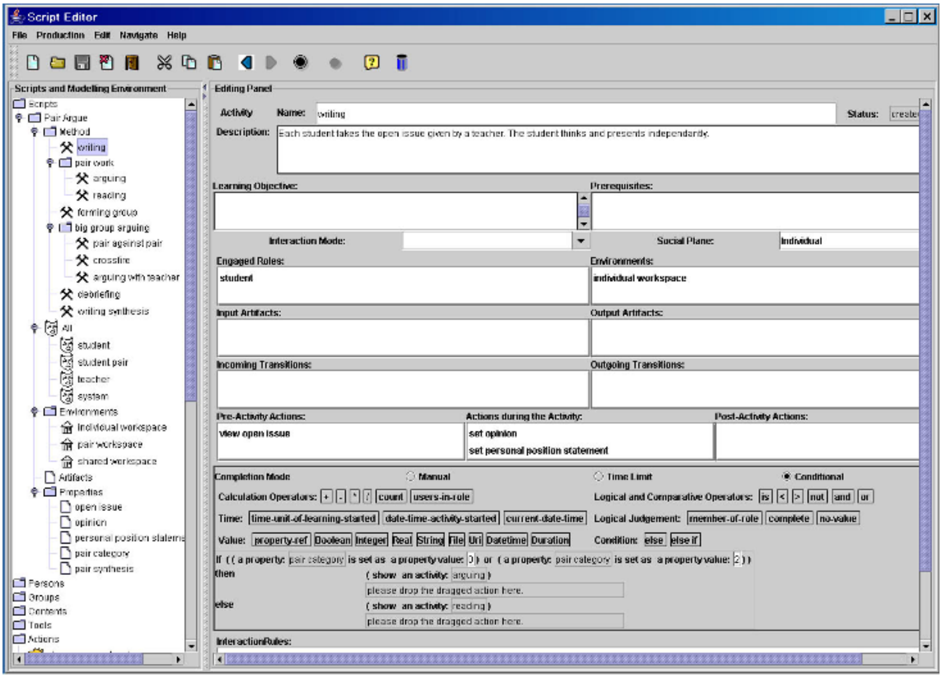
However, the element operations are insufficient to support modeling adaptation logics and adaptation actions for groupalization. As an extension, we introduce new operations: Examples of extended get()-like element operations are users-in-group and roles-taken-by-group. The examples of extended set()-like element operations are assign-group-to-role and remove-user-from-group. In addition some de-/construction operations are added such as create-role and delete-group. Also, we add declaration mechanisms to define complicated expressions and actions. An expression declaration primarily consists of two parts. The first part is a representation of internal operational structure based on the extended IMS LD {expression} schema group and element operations. The second part is a user-friendly representation of the expression with a set of parameters. Correspondingly, an action declaration is defined in the same way. We add 'collection' data type and loop control structure to support complicated declarations. A declaration is indeed a procedure writing in the process modeling language, which can be interpreted into executable programming language code based on element operations. After being defined, a declaration can be saved in the modeling environment, and it can be used to define other high-level declarations as well. Then, an expression or an action can be defined by referring to the declaration with parameters without concerning about the internal operational structure. Therefore, we can support learning designers to specify rich adaptation logics and adaptation actions.

In addition, IMS LD adds the capability for a learning designer to specify sending messages and setting new activities based on certain events at level C. We extend such a notification mechanism by introducing the concept of interaction rules. An interaction rule is specified by defining a condition, an agent (e.g., a user, a group, or a role), a permission right, and a set of actions. It will be triggered by certain events and informs an agent to perform actions. The run-time system will provide appropriate user interface for the associated users to perform an action directly rather than just receiving a notification via an email. This mechanism can be used to support human involved adaptation. For example, in the example model, when it fails to form high-level groups, the run-time system should update the user interface of the teacher's client for the teacher to adjust activities for the remaining pairs.

4. An Authoring Tool

As mentioned before, we developed a CSCL scripting language [7]. One objective of the language is to facilitate formalization of adaptive learning design for groupalization. Based on the language, we developed an authoring tool, called CoSMoS (for "Collaboration Script Modeling System"). It can help designers to understand and specify learning designs (or CSCL scripts), which can be translated from/into XML-formatted files automatically by the tool. The adaptive learning design files can be used by a run-time system to adapt the course during the execution by adjusting the activities to groups' characteristics and by providing appropriate user interfaces for the group members.

Figure 2: The User Interface of COSMOS and the Definition of the Example Model



Activity Name: writing
Status: create

Description: Each student takes the open issue given by a teacher. The student thinks and presents independardly.

Learning Objective:

Prerequisites:

Interaction Mode:

Social Plane: Individual

Engaged Roles: student

Environments: individual workspace

Input Artifacts:

Output Artifacts:

Incoming Transitions:

Outgoing Transitions:

Pre-Activity Actions: view open issue

Actions during the Activity: set opinion
set personal position statement

Post-Activity Actions:

Completion Mode: ○ Manual ○ Time Limit ● Conditional

```
If ( ( a property: pair category is set as  a property value: 0 ) or  ( a property: pair category is set as  a property value: 2 ) )
then
                    ( show  an activity: arguing )
                    please drop the dragged action here.
else
                    ( show  an activity: reading )
                    please drop the dragged action here.
```

The user interface of CoSMoS is shown in Figure 2. The window of the tool consists of a toolbar and two panels. The left panel is used to define the structure of adaptive learning designs and the right panel is used to create detailed designs for the selected process element. We have applied the tool to defining several CSCL scripts and so far we found that the tool and the underlying CSCL scripting language provide sufficient mechanisms to model adaptive learning designs. Figure 2 shows a definition of the example model described in the section 1. In the structure panel, the 'pair argue' script is shown as a tree. Expression declaration nodes, action declaration nodes, and other modeling environment components are listed below the script nodes. The editing panel shows the specification of the first activity. In this panel, an adaptive rule is specified by defining a conditional expression including a local group property "pair category" and two alternative activities. The run-time system will adapt activities according to such a definition.

5. Conclusions and Future Work

The objective of this paper was to outline a framework for an education modeling language that integrates new elements for supporting groupalized learning. The proposed framework is based upon IMS LD, which provides mechanisms to specify adaptive learning designs for personalization. After introducing the group element and adding element operations, declarations, and interaction rules, our CSCL scripting language can meet requirements identified through an analysis of a scenario. The preliminary 'proof of concepts' of the CSCL scripting language was given in using our authoring tool CoSMoS. Preliminary tests show that adaptive learning designs for groupalization can be formalized by using CoSMoS.

As described our approach facilitates the specification of learning designs derived from pedagogical principles without representing deeper reasons for the one or the other choice of method or interaction pattern. Evidently, existing work on intelligent group formation and the management of learning groups (as, e.g., described in [4]) could extend this approach with "expert knowledge".

The validation results of the real experiments will have to look into more detail whether the approach taken is successful. In particular, experiments should be conducted on the corresponding run-time systems to demonstrate the adaptability during the execution of adaptive learning designs. We have confidence in this approach, because IMS LD can support personalization. Therefore, our next step is to develop a compatible execution environment that can interpret CSCL scripts and provide run-time supports. Meanwhile, we will develop more adaptive learning designs to facilitate groupalized learning.

References

[1] Brusilovsky, P. (2001) Adaptive Hypermedia. UM and User Adapted Interaction, vol. 11(1/2), 2001, pp. 87-110.

[2] Hummel, H. G. K., Manderveld, J. M., Tattersall, C.,& Koper, E. J. R. (2004). Educational Modeling Language: new challenges for instructional re-usability and personalized learning. International Journal of Learning Technology, vol.1, No.1, pp.111-126.

[3] IMS Learning Design Best Practice and Implementation Guide; IMS Learning Design Information Model; IMS Learning Design XML Binding. IMS Global Learning Consortium. Version 1.0 Final Specification, Revision 20.01.03. Download at http://www.imsglobal.org

[4] Inaba, A., Tamura, T., Okhubo, R., Ikeda, M., Mizoguchi, R., Toyoda, J. (2001). Design and analysis of learner interaction based on collaborative learning ontology. Proc. of Euro-CSCL, Maastricht (NL), March 2001, pp. 308-315.

[5] Koper, E.J.R. & Olivier. B. (2004). Representing the Learning Design of Units of Learning. Educational Technology & Society. 7(3), pp.97-111.

[6] Koper, E.J.R. & Tattersall, C. (Eds), Learning Design: A Handbook on Modelling and Delivering Networked Education and Training, Springer, Berlin, 2005.

[7] Miao, Y., Hoeksema, K., Hoppe, U. Harrer, A. (in press). CSCL Scripts: Modeling Features and Potential Use. Proceedings of Computer Supported Collaborative Learning (CSCL) conference 2005.

[8] Santos, O.C., Boticario, J.G., and Barrera, C. (2004). Authoring a collaborative task extending the IMS-LD to be performed in a standard-based adaptive Learning Management System called aLFanet. In post-proceeding volume of the International Conference on Web Engineering (ICWE'04).

[9] Santos, O.C., Boticario, J.G., and Barrera, C. (2004). Artificial Intelligence and standards to build an adaptive Learning System. Proceedings of the 14th International Conference on Computer Theory and Applications (ICCTA'2004). Ed. IEEE, 2004.

[10] Van Rosmalen, P., Brouns, F., Tattersall, C.,Vogten, H. Van Bruggen, J, Sloep, P., & Koper, E.J.R. (in press). Towards an Open Framework for Adaptive, Agent-supported e-learning. International Journal of Continuing Engineering Education. Available at http://hdl.handle.net/1820/76

[11] Weber, G. and Brusilovsky, P. (2001). ELM-ART: An adaptive versatile system for Web-based instruction" International Journal of Artificial Intelligence in Education, vol. 12(4), Special Issue on Adaptive and Intelligent Web-based Educational Systems, pp.351-384.

[12] Wenger, E. Artificial Intelligence and Tutoring System. Morgan Kaufmann, 1987.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

On the Prospects of Intelligent Collaborative E-learning Systems

Miikka Miettinen [a,1] Jaakko Kurhila [b] and Henry Tirri [c]
[a] *Helsinki Institute for Information Technology, Finland*
[b] *Department of Computer Science, University of Helsinki, Finland*
[c] *Nokia Research Center, Nokia Group, Finland*

Abstract. Collaborative learning is question-driven and open-ended by nature. Many of the techniques developed for intelligent tutoring are applicable only in more structured settings, but fortunately there are other interesting opportunities to explore. In this paper we introduce a system called OurWeb, and use it as an exemplar framework for demonstrating some of these opportunities. We claim that effective participation in distributed and self-organizing collaboration requires sufficient awareness of the resources and dynamics of the community. The feasibility of implementing certain features of this kind is evaluated based on data from two university level courses.

Keywords. collaborative learning environments, collaborative learning, collaborative annotation, awareness, social navigation, information retrieval

1. Introduction

The objective of intelligent e-learning systems, as it is typically conceived, is to provide highly structured lessons that are to a large extent under automated control. Within this framework, the intelligence of the system often appears in the form of adaptive sequencing or personalization of the course material, adaptive guidance for navigation, or interactive problem solving support. All of these methods work the best in well-structured domains, and rely heavily on a fixed collection of pre-made course material.

While the prevailing approach has arguably proved to be appropriate in several contexts, there are good reasons to extend the perspective to other essential ways of learning. On the one hand, the theoretical assumptions implicit in the instruction method have received substantiated criticism. Learning has been claimed to be primarily a matter of participation [1] or collaborative knowledge building [2] rather than direct assimilation of facts from an authoritative source. The critics have suspected that excessive guidance places the students in a passive role, hampers the development of metacognitive skills, and results in an instructional setting that is too simplified and restricted to facilitate real-world problem solving [3,4,5]. These claims may or may not be justified, but in any

[1] Correspondence to: Miikka Miettinen, Helsinki Institute for Information Technology, P.O.Box 9800 FIN-02015 TKK, Finland; Tel.: +358 9 451 8123; E-mail: miikka.miettinen@hiit.fi.

This work was supported in part by the Academy of Finland under the Prima and Prose projects.

case they highlight the fact that some important aspects of learning do not fit well in the present framework.

On the other hand, collaborative learning has become a fairly common way of organizing education, and attempts to develop better tools for its particular needs are motivated in their own right. However, the needs turn out to be quite different from the ones that intelligent e-learning systems typically try to address. The collaborative learning process is highly unstructured and open-ended, and the activities of an individual student must be considered in a broader context. As a result, the most interesting opportunities to develop intelligent functionality are related to facilitating collaboration rather than adapting the learning material.

The next section introduces a system called OurWeb, which demonstrates the principles of collaborative learning and provides a suitable exemplar framework for the rest of the paper. In section 3 we present some general ideas of advanced features that might support the collaborative learning process, and continue with a preliminary feasibility study in section 4. Section 5 concludes with some general reflections of the issues involved.

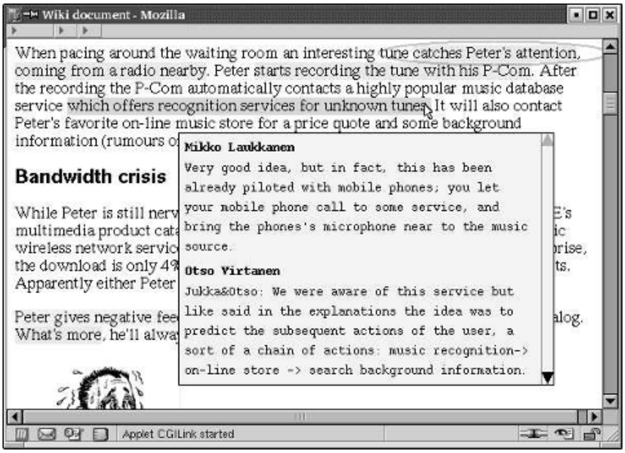
2. Collaborative Learning with OurWeb

Collaborative learning takes place within the framework of joint activities. Rather than trying to master a fixed set of topics determined by the instructor, the students are engaged in an open-ended effort to advance their collective understanding [4]. Division of work and specialization are seen as opportunities, and the students are encouraged to rely on each other as sources of information and assistance. Genuine participation taking place in a meaningful social context is claimed to make learning a matter of personal development and result in deep intrinsic motivation [1]. In addition, interactions among the students facilitate learning directly by encouraging them to explain the subject matter to each other and revealing in a constructive way the inconsistencies and limitations in their knowledge [6].

OurWeb is an integrated set of tools for collaborative learning. The most essential principles underlying its design are *openness* and *transparency*. By openness we mean that the students should be enabled to utilize any available information sources with as few restrictions as possible. Transparency is pursued by attempting to provide tools that fuse seamlessly into the activities of the students, allowing them to benefit from the work of each other and participate in meaningful ways.

The OurWeb server acts as a proxy between the user's browser and the Web, capable of augmenting any page with additional content and functionality. Most features are located in a custom *popup menu*, which is opened with the right mouse button. Some of the menu items are used for manipulating the visible page and others for navigating between various parts of the system. This kind of a minimalist user interface is natural and appropriate when providing unrestricted access to heterogeneous Web pages.

OurWeb provides a shared *document pool*, which serves as a repository for both external resources and the students' own work. Any potentially useful Web page can be linked to the repository with the popup menu. The user simply opens the menu with the right mouse button and chooses the option labeled "Add to document pool". As a result, the document becomes visible to everyone on the various index pages and the internal



When pacing around the waiting room an interesting tune catches Peter's attention, coming from a radio nearby. Peter starts recording the tune with his P-Com. After the recording the P-Com automatically contacts a highly popular music database service which offers recognition services for unknown tunes. It will also contact Peter's favorite on-line music store for a price quote and some background information (rumours o

Bandwidth crisis

While Peter is still nerv … E's
multimedia product cat … ic
wireless network servic … prise,
the download is only 4% … ts.
Apparently either Peter

Peter gives negative fee … alog.
What's more, he'll alwa

Mikko Laukkanen

Very good idea, but in fact, this has been already piloted with mobile phones; you let your mobile phone call to some service, and bring the phones's microphone near to the music source.

Otso Virtanen

Jukka&Otso: We were aware of this service but like said in the explanations the idea was to predict the subsequent actions of the user, a sort of a chain of actions: music recognition-> on-line store -> search background information.

Figure 1. Comments in OurWeb.

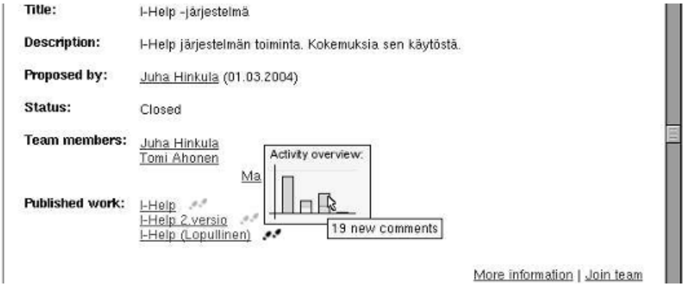
search engine, and the full functionality of OurWeb (including e.g. annotations) can be applied to processing the contents effectively.

In collaborative learning, different groups of students are typically working on different topics, and the groups are organized by the students themselves instead of being assigned by the instructor. OurWeb supports the process by enabling the students to publish their ideas and suggestions as *projects*. The initial proposal consists of a title and a short description of the content, along with plans and schedules for organizing the effort in practice. Interested people can get involved by simply clicking a link labeled "Join team". All ideas do not normally create sufficient interest, and the person who made the suggestion does not necessarily need to participate as an active team member. We want to avoid creating unnecessary barriers to collaboration, and encourage participation in all forms.

During the course of a project the team members are engaged in collaborative *process writing*. The idea is to produce a document incrementally, gathering feedback and ideas from the others along the way. In addition to supporting the work of each individual group, this enables cross-fertilization of ideas and fosters the sense of being part of a larger community.

OurWeb contains an integrated *Wiki*, which the groups use as a document editor. A Wiki (or WikiWikiWeb) is a tool for collaborative authoring of Web pages with an ordinary Web browser and a simple markup language [7]. At any point in time, the team has an internal "working copy" of the document being written. Intermediate versions can be published in the document pool with one mouse click, and are essentially snapshots of the continuously evolving document. The groups are encouraged to publish the first drafts already at the early stages of the work in order to get feedback and create opportunities for collaboration.

The primary means of collaboration are annotations and threaded discussions. Two different types of annotations are supported: *highlights* and *comments*. Highlights can be applied to marking important parts of the text, analogously to the way many people underline text on paper. In practice, adding a highlight involves selecting the text with the mouse, right-clicking the mouse to make the popup menu visible, and choosing the "Highlight" option from the menu. Comments are added the same way, except that the



Title: I-Help -järjestelmä

Description: I-Help järjestelmän toiminta. Kokemuksia sen käytöstä.

Proposed by: Juha Hinkula (01.03.2004)

Status: Closed

Team members: Juha Hinkula
Tomi Ahonen

Published work: I-Help
I-Help 2.versio
I-Help (Lopullinen)

More information | Join team

Figure 2. Footprint information of a project's published document version.

user types the input in a popup window. A comment appears as a tooltip when the mouse pointer is placed on top of the commented text fragment (see Figure 1). If several comments are attached to the same text, they appear one after the other as a dialogue. Longer reflections and remarks that may not be associated with any single passage of text can be posted in a *threaded discussion* located at the bottom of the page.

The annotation and discussion facilities of OurWeb allow the community to engage in *artifact-centered discourse* [8], in which the contributions appear in the immediate proximity of the relevant information. This has turned out to be very useful in practice. We have observed that especially comments are used extensively for short exchanges of feedback and ideas that would probably never have taken place in a detached discussion forum.

The number of documents in the document pool can grow large, and it is useful to provide several alternative views to the contents. These include e.g. lists organized by topic and the navigation history of the user, and a selection of documents that have recently received attention from the community. The system also contains an *internal search engine* covering the document pool as well as the annotations and threaded discussions. Google can be used through the OurWeb server for searching the entire Web.

Each link appearing on the index pages is followed by a *footprint icon*, which is either black or gray, depending on whether or not the document has received new activity since the user's previous visit (see Figure 2). When the user places the mouse pointer over the icon, a bar chart appears showing the relative amount of reading, highlighting, commenting, and threaded discussion activity associated with the document.

Other features of OurWeb include personalized desktop, automatic marking of new comments, and an interface for sending e-mail. The desktop serves as the entry point to the system, and contains recommended links to documents and discussion messages along with announcements from the instructor. Marking of new comments makes it easier to follow the gradual progress of asynchronous collaboration. The marks appear as ovals or lines around the commented text fragments (see the upper right corner of Figure 1). Finally, e-mail messages can be sent conveniently to an individual user or everyone in a particular project team by clicking links appearing in the project list.

3. Suggested Features for Intelligent Collaborative E-learning Systems

The "intelligent" functionality that is feasible and appropriate in the collaborative setting has to be quite different from a conventional intelligent e-learning system. The students are engaged in question-driven and open-ended inquiry, which would be very difficult to augment with automated guidance and problem solving support. In addition, it is not obvious that such facilities would be appropriate, even if they were feasible to implement. Identifying fruitful lines of inquiry and exchanging explanations in peer groups are essential elements of collaborative learning that should not be transferred away from the students.

Therefore, we propose a different approach. Rather than trying to guide the students directly, the system should support their activities with various kinds of supplementary information. Effective participation in distributed and self-organizing collaboration requires sufficient *awareness* of the resources and dynamics of the community. A suitable role for the system is to try to provide the right information at the right time, while the interpretation of the information and the associated decision making are best left to the user.

It seems plausible that several key activities involved in collaborative learning could be supported by better awareness. In this section we identify some relevant objectives and present general ideas of the additional functionality that would be needed for achieving them. The next section presents some data gathered from OurWeb in an attempt to assess in more detail the need for automated recommendation of collaboration opportunities.

3.1. Facilitating effective utilization of background material

At the age of the Internet, collaborative learning often happens at the edge of information overload. For almost any question the students might choose to examine, there is an endless supply of partially overlapping resources with additional details. The problem is not primarily technical by nature, but better tools could make it easier to locate relevant information and utilize the work of the others.

One obvious approach is to try to develop better facilities for information retrieval. In addition to the keyword search included in the current version of OurWeb, we have done some preliminary experimentation with *proactive search*. The idea is to observe the navigation and scrolling patterns of the user, and generate queries automatically to provide additional links to potentially relevant pages. Unlike the user, the search engine has a *global view* of the available contents and could (at least in principle) identify semantic relations between disparate sources of information. If successful, this would provide the user with improved awareness of the available contents, and reduce the cognitive demands involved in reading and constructing explicit queries at the same time.

Potentially relevant material can also be highlighted by presenting *recommended links*. In the absence of an explicit domain model, such recommendations are typically based on content-based or collaborative filtering. Both techniques rely on the notion of a *user profile*, which is assumed to be stable over long periods of time. In the present context this assumption is clearly invalid, because the usefulness of a document changes dynamically both as a result of learning and depending on the task that the student is working on at a particular moment.

Therefore, a better approach is to resist the temptation to give explicit recommendations, and focus on supporting the users' own judgment. For example, the kind of data underlying the "footprints" of OurWeb could be used as input for collaborative filtering, but presenting it directly to the users in summarized form is much more transparent and informative. Other examples of supporting cooperative processing of background material include OurWeb's shared document pool and annotations. Enabling the students to rely on the work of each other allows them to achieve a higher level of understanding than what would be possible if the same routines had to be repeated by each individual.

3.2. Making collaboration opportunities apparent

Informing the students about the activities of each other would also facilitate direct interactions. The shared workspace provides many *opportunities* for collaboration, and active encouragement from the system could make a significant difference in the engagement of the students. Ideally, the suggestions would be personalized and context-sensitive, adapted both to the needs of the individual and the overall status of the community. High precision would not be vital, however. Even if the suggestions were not especially pertinent, they might increase the amount of collaboration just by encouraging people to contact each other.

In order to form groups, the students need to be aware of the interests of each other. A suitable way of supporting such awareness would be to augment documents with information about people who have been actively utilizing them [9]. This would enable the students to identify potential collaboration partners when coming across interesting material.

When the groups are engaged in process writing, it is beneficial for their motivation and efficiency to get timely feedback. The system could encourage this by providing explicit notifications to potential reviewers. On the one hand, it would be appropriate to inform them whenever a new document version is published for review. Avoiding delays would ensure that the comments are valid and taken into account, as the document is often under continuous revision. On the other hand, the authors and the reviewers typically engage in asynchronous discussions, the status of which could be monitored and summarized automatically by the system. This would also help to eliminate delays by providing the users with better awareness of the progress of the discussions.

Real-time awareness of the presence of the others would facilitate *peripheral monitoring* of the workspace. When supplemented with synchronous communication tools such as chat and instant messaging, it would enable the students to engage in *spontaneous collaboration* motivated by momentary needs. This is claimed to be particularly useful in collaborative writing, which is characterized by frequent switches between independent work and focused group consideration of individual details [10].

3.3. Supporting coordination of group work

Effective group work also requires awareness of the activities of the other participants. Individual students need to coordinate the content and timing of their contributions with each other, and keep their efforts aligned with the overall objectives of the group. It is typical that the activities are reorganized repeatedly as new ideas and better understanding emerge [11]. Although continuous awareness can be maintained by means of explicit communication, utilizing data that accumulates automatically as a side product of the activities decreases the amount of routine communication and helps to eliminate unnecessary delays.

Different stages of the work call for different degrees of collaboration. Better awareness of the progress would enable flexible shifts between close and loose collaboration and make the interactions more fluid and natural [10]. Interestingly, it would also provide a basis for shared norms and conventions. The availability of relevant information would remove certain kinds of ignorance from the set of legitimate excuses, and foster stronger commitment the joint effort [9].

4. Feasibility Study

4.1. Study setting

Our empirical study assessed the need and feasibility of implementing automated recommendation of collaboration opportunities. We focused on three particular objectives:

1. *Supporting group formation by identifying students with shared interests.* As suggested in the previous section, a suitable way of supporting group formation might be to augment documents with information about people who have been actively utilizing them. The prerequisites for this would be the emergence of interest profiles from the activity patterns of the students, and sufficient overlap in the navigation of students with similar profiles.
2. *Increasing the timeliness of feedback.* The system could try to increase the fluidity of the review process by providing explicit notifications to potential reviewers. However, it would be useful to know specifically what kind of delays actually occur in the absence of this functionality.
3. *Providing real-time awareness of the presence of the others.* In order to cater for spontaneous collaboration, the system could inform the users about the presence and activities of each other. This is feasible only to the extent that there are several users logged in the system simultaneously.

The data was acquired from two university courses that employed the current version of OurWeb. During the first course titled “Computer Uses in Education”, 17 students were working on self-organized projects over a period of 10 weeks. The arrangements were extremely flexible, allowing the students to participate in projects of their own choice with roles and schedules negotiated among themselves. The second course involved doing a written and oral presentation on a free topic related to “Web Communities”. There were 16 students, and the work was done over a period of 7 weeks. The students of both courses were predominantly male and computer science majors.

4.2. Results

When a document is added to the shared document pool of OurWeb, it is assigned manually to one or more *topics*. As the first part of our analysis, we wanted to see if it would be possible to support group formation by identifying students with shared interests. We looked at the distribution of the students’ reading time with respect to the topics during the one week period preceding the formation of each group. Clear differences in the reading activity of the students were found. In 45% of the cases a single topic accounted for 50% or more of the student’s total reading time. There was also sufficient overlap in visits to individual documents. For example, for those with a clear interest profile on average 3 other students with the same profile had also visited a particular document associated with the dominant topic. Therefore, it seems that the suggested type of support for group formation could have been provided in practice.

There is also room for improvement in the timeliness of the feedback received by the project teams. On average only 42% of the feedback was received during the first two days after the publication of a draft, and 36% was received after 5 days or more. Turn taking in comment chains and discussion threads had an average delay of 38 hours.

Opportunities for synchronous interaction would have been limited. On average there were just 2.1 users online simultaneously, and the number went rarely above 5. Therefore, it seems that at least in small courses like ours the value of real-time awareness is questionable.

5. Conclusions

Collaborative learning is question-driven and open-ended by nature. Many of the techniques developed for intelligent tutoring are applicable only in more structured settings, but there are other interesting opportunities to explore. In this paper we suggested that trying to provide awareness of potentially relevant activities and resources is an appropriate direction for these explorations, and took some preliminary steps towards the implementation of such tools.

References

[1] Wenger, E. (1999). Communities of Practice: Learning, Meaning and Identity. Cambridge, UK: Cambridge University Press.

[2] Bereiter, C. (2002). Education and mind in the knowledge age. Mahwah, NJ: Lawrence Erlbaum Associates.

[3] Bredo, E. (1993). Reflections on the intelligence of ITSs: a response to Clancey's "Guidon-manage revisited". International Journal of Artificial Intelligence in Education, 4, 35-40.

[4] Scardamalia, M. & Bereiter, C. (1994). Computer support for knowledge-building communities. The Journal of the Learning Sciences, 3(3), 265-283.

[5] Deek, F.P. & McHugh, J. (1998). A Review and Analysis of Tools for Learning Programming. In Proceedings of the ED-MEDIA World Conference on Educational Multimedia, Hypermedia and Telecommunications, pages 251-256. AACE.

[6] Hatano, G. & Inagaki, K. (1993). Desituating cognition through the construction of conceptual knowledge. In Light, P. and Butterworth, G. (eds.), Context and Cognition: Ways of Learning and Knowing, 115-133. Lawrence Erlbaum Associates.

[7] Leuf, B. & Cunningham W. (2001). The Wiki Way: Quick Collaboration on the Web. Addison-Wesley Longmann.

[8] Suthers, D. & Xu, J. (2002). Kukakuka: An Online Environment for Artifact-Centered Discourse. In Proceedings of the Eleventh World Wide Web Conference (WWW 2002), 472-480.

[9] Erickson, T. & Kellogg, W.A. (2000). Social translucence: An approach to designing systems that support social processes. ACM Transactions on Computer-Human Interaction, 7(1), 59-83.

[10] Dourish, P. & Bellotti, V. (1992). Awareness and coordination in shared workspaces. In Proceedings of the 1992 ACM Conference on Computer-Supported Cooperative Work, pages 107-114. ACM Press.

[11] Gutwin, C. & Greenberg, S. (2002). A Descriptive Framework of Workspace Awareness for Real-Time Groupware. Computer Supported Cooperative Work, 11(3), 411-446.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

COFALE: An Adaptive Learning Environment Supporting Cognitive Flexibility

Vu Minh Chieu and Elie Milgrom
Department of Computing Science and Engineering
Université catholique de Louvain
2, Place Sainte-Barbe, B-1348 Louvain-la-Neuve, Belgium
vmc@info.ucl.ac.be, em@info.ucl.ac.be

Abstract: Constructivism is a learning theory that states that people learn best when they actively construct their own knowledge. Various forms of "constructivist" learning systems have been proposed in recent years. According to our analysis, those systems exhibit only a few constructivist principles, and few of them support adaptation to different kinds of students.

Our research aims to design truly constructivist and adaptive learning environments. Our approach is based on a set of operational criteria for certain aspects of constructivism: We use these criteria both as guidelines for designing our learning system and for evaluating the conformity of our learning system with constructivist principles.

One of the facets often mentioned as being strongly relevant to constructivism is cognitive flexibility. This paper presents COFALE, a domain-independent adaptive e-Learning platform that supports cognitive flexibility, and an example of its use.

Introduction, Background, and Context

"Constructivist" learning systems

Constructivism, as defined by Santrock [16], is an educational approach that "emphasizes that individuals learn best when they actively construct knowledge and understanding" (p. 318). Bruner [4] introduces the following example of constructivist learning:

> The concept of prime numbers appears to be more readily grasped when the child, through construction, discovers that certain handfuls of beans cannot be laid out in completed rows and columns. Such quantities have either to be laid out in a single file or in an incomplete row-column design in which there is always one extra or one too few to fill the pattern. These patterns, the child learns, happen to be called prime. It is easy for the child to go from this step to the recognition that a multiple table, so called, is a record sheet of quantities in completed multiple rows and columns. Here is factoring, multiplication and primes in a construction that can be visualized.

In recent years, constructivist beliefs and practices have been widely adopted, as evidenced by the appearance of several "constructivist" learning systems [13]. Many researchers accept the central assumption of constructivism as stated by Santrock; however, they derive different pedagogical implications from the same basic principles. Driscoll [8], for instance, identifies five major facets of constructivism related to instructional design: (1) reasoning, critical thinking, and problem solving; (2) retention, understanding, and use; (3) cognitive flexibility; (4) self-regulation; and (5) mindful reflection and epistemic flexibility. Existing learning systems exhibit only at most a few constructivist principles from this list.

In earlier work [6], we have defined a set of operational criteria for cognitive flexibility (**CF**) and we have used these criteria to evaluate systems such that SimQuest [10], Moodle [7], KBS [12], claimed by their authors to be "constructivist". We discovered that these systems support only a small part of the various pedagogical principles underlying CF.

A second characteristic we would like to see implemented in computer-based learning systems is adaptability, i.e. the ability to provide a learning experience that is tailored to the needs of the individual learner [5]. Except for KBS, none of the "constructivist" systems we looked at effectively implements adaptation support.

Contributions

In this paper, we show how to exploit available learning technologies to design adaptive learning environments that truly facilitate and stimulate CF, one of the important facets of constructivism often mentioned by constructivist researchers. In support of our claim, we present a new e-Learning platform, named COFALE, in which we provide every learner with personalized learning situations that extensively support CF. We illustrate the use of COFALE to support CF in a problem area presented in the next paragraph, the learning of the concept of recursion. Our approach is based on the set of operational criteria for CF [6] used as guidelines and means of validation for the design of COFALE.

Context for the examples

The concept of recursion is very important in computing science [3]. Many consider that both teaching and learning recursion are difficult because of three main reasons [1]: (1) the concept is unfamiliar (students are induced to proceed by analogy from examples); (2) the concept is complex (it is hard for students to transfer from a pattern of recursion to a new one); and (3) interference may arise from knowledge of other methods of solution (e.g. iteration).

Structure of the paper

Sections 1 and 2 introduce necessary background on CF and adaptability; section 3 shows how a course designer might use COFALE to devise adaptive learning situations leading to CF; section 4 presents our analysis of COFALE and related work; the last section presents our conclusions and future work.

1. Cognitive Flexibility

According to Spiro and Jehng [17], CF is “the ability to spontaneously restructure one’s knowledge, in many ways, in adaptive response to radically changing situational demands” (p. 165). We propose here a simple example to clarify this concept (Spiro’s paper contains several other ones): A child, through personal experience and interactions with peers, parents, and teachers, develops the ability to (re)structure its own knowledge, in many ways, to be able to derive the meanings of the same word in different contexts. For example, given the sentence “I watched the bat flitting through the trees”, the child considers the word “bat” as a noun, then as an animal, then as the actual meaning of this word. Given another sentence “I hope I can bat a home run”, the child will consider the word “bat” as an action verb, then as the actual meaning of this verb.

Driscoll [8] identifies two principal learning conditions that stimulate CF: (1) multiple modes of learning (i.e., multiple representations of contents, multiple ways and methods for exploring contents); and (2) multiple perspectives on learning (i.e., expression, confrontation, and treatment of multiple points of view).

Chieu and colleagues [6] transformed the pedagogical principles underlying the previous two learning conditions for CF into operational criteria. They define an operational criterion for CF to be “a test that allows a straightforward decision about whether or not a learning situation [reflects] the pedagogical principles underlying CF”. They first examine many existing learning systems and identify three main components of learning systems: (1) learning contents (e.g. concept definitions); (2) pedagogical devices (e.g. tools provided for learners for exploring learning contents); and (3) human interactions (e.g. means for engaging tutors and learners in exchanges). Then, in each of the three learning components and for each of the two learning conditions for CF, they propose criteria that can be applied for checking the presence of the learning condition in the learning component (Table 1).

Table 1. Operational Criteria for CF by Chieu et al. [6] (MM = Multiple Modes, MP = Multiple Perspectives)

Learning Contents
MM1: *The same learning content presenting concepts and their relationships is represented in different forms (text, images, audio, video, simulations, …).* **MP1:** *The same abstract concept is explained, used, and applied systematically with other concepts in a diversity of examples of use, exercises, and case studies in complex, realistic, and relevant situations.*
Pedagogical Devices
MM2: *Learners are encouraged to study the same abstract concept for different purposes, at different times, by different methods including different activities (reading, exploring, knowledge reorganization, etc.).* **MP2:** *When facing a new concept, learners are encouraged to explore the relationships between this concept and other ones as far as possible in complex, realistic, and relevant situations.* **MP3:** *When facing a new concept, learners are encouraged to explore different interpretations of this concept (by other authors and by peers), to express their personal point of view on the new concept, and to give feedback on the points of view of other people.* **MP4:** *When facing a new concept, learners are encouraged to examine, analyze, and synthesize a diversity of points of view on the new concept.*
Human Interactions
MM3: *The number of participants, the type of participant (learner, tutor, expert, etc.), the communication tools (e-mail, mailing lists, face to face, chat room, video conferencing, etc.), and the location (in the classroom, on campus, anywhere in the world, etc.) are varied.* **MP5:** *During the discussion, learners are encouraged to diversify – as far as possible – the different points of view about the topic discussed.*

This set of criteria was used to analyze existing systems and in the design of COFALE. In section 3.1, we show that all the criteria in Table 1 are satisfied by means of learning situations proposed to the learner for the example handled in COFALE.

2. Mental Models and Adaptability

2.1 Mental Models

In a constructivist point of view, each learner possesses a mental model (i.e. a mental representation or knowledge structure) about a concept or a situation at any point in time. The purpose of learning is to have the mental model get closer and closer to that subsumed by the learning objectives. Through personal experience, the learner may undergo a certain number of cognitive changes and then develop a higher mental model. For instance, a beginner could start with a "novice" model on a given subject and gradually evolve toward an “expert” model through his or her learning. One of the major roles of the designer of a "course" is thus to provide the learner with appropriate learning conditions to facilitate the learner’s process of knowledge construction and transformation [16].

Several researchers [1, 3] have interviewed students and analyzed students’ tests on the subject of *recursion*. They distinguish four approaches that students try to apply to generate recursive solutions to a given problem:

- *Loop model:* “Novice” learners, when constructing a recursive solution, try to adapt some part of an iterative structure, e.g. the updating of loop index variables, in order to achieve recursion.
- *Syntactic model*: Learners consider recursion as a template consisting of a base case and a recursive part. Although they may not fully understand the functionality of the recursive part, they are able to solve simple problems by filling the condition part and the action part of the base case and the recursive part.
- *Analytic model*: Learners consider recursion as a problem-solving technique. They analyze diverse cases of a given problem; then, for each case, they determine input conditions and output actions; finally, they write recursive code.
- *Analysis-synthesis model*: “Expert” learners, in addition to the ability implied by the analytic model, are able to apply the DCG (Divide, Conquer, and Glue) strategy to solve problems recursively: They break a large problem into one or more sub-problems that are identical in structure to the original problem and somewhat simpler to solve.

From our point of view, each of these approaches may be seen as defining the mental model of a learner getting acquainted with the concept and applications of recursion.

2.2 Adaptability

Brusilovsky [5] presents four main techniques for implementing adaptability: (1) presentation of learning contents (e.g. define which contents are appropriate to a specific learner at any given time); (2) presentation of pedagogical devices (e.g. define which learning activities are appropriate to a specific learner); (3) communication support (e.g. identify which peers are appropriate to help a specific learner); and (4) problem-solving support (e.g. give appropriate feedback during the problem-solving process of a specific learner).

Only the first three techniques presented by Brusilovsky are domain-independent; section 3.2 shows how we apply them in COFALE, in a manner consistent with the constructivist point of view presented earlier, to adapt the learning contents, pedagogical devices, and communication support to the different kinds of learners identified previously.

3. COFALE as a Learning Environment

COFALE is an adaptive learning environment supporting CF; COFALE is based on ATutor, an open-source, Web-based learning content management system (LCMS) designed and maintained by the ATRC group [2]. For the purpose of the discussion, we shall assume that a “novice” learner (Bob), familiar with "traditional" programming (say, in the Java language) and thus with the concept of iterations, uses COFALE to learn recursion (i.e. to develop the ability to solve problems recursively); a tutor and a number of other learners (peers) also participate in the same learning experience. In section 3.1, we show for each criterion presented in Table 1 for CF, how the course designer uses COFALE to present Bob with learning situations satisfying the corresponding criterion. Section 3.2 explains how the course designer uses COFALE to provide Bob and his peers with adaptation support.

3.1 Learning with Support for Cognitive Flexibility

Bob needs to develop his capacity to implement recursive solutions for a variety of problems. After reading the definition and examples of the main concepts (recursion, recursive algorithms, and recursive methods), Bob is encouraged to explore a situation about arithmetic expressions (Figure 1). We show below, in the presentation for criterion MM2, how Bob is encouraged, in COFALE, to explore situations.

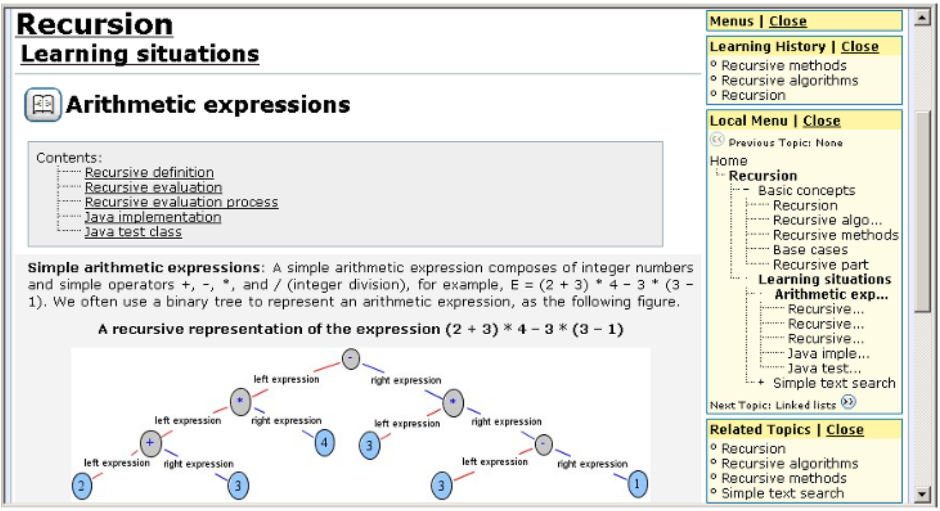
Criterion MM1. In the arithmetic expressions situation, the course designer induces Bob to examine multiple representations of recursion through the “Local Menu” seen on the right hand side of Figure 1: a textual definition, two simulations, and a Java implementation.

To satisfy criterion MM1, the course designer has made multiple representations available for recursion: A combination of text, images, and simulations helps Bob grasp diverse aspects of recursion better than a single text does.

Criterion MP1. After exploring the first situation, Bob is encouraged to explore the second one: “Simple text search”, seen at the bottom of the menu “Related Topics” offered by ATutor, thus also by COFALE (Figure 1). In this situation, Bob sees how to apply recursion to represent a text (i.e. a list of words) as a linked list and how to look up a phrase in a document.

In COFALE, we explicitly encourage the course designer to prepare several situations to help Bob understand how to apply the concept of recursion in different contexts. Arithmetic expressions explain the use of recursion in binary trees in a natural way and simple text search explains the use of recursion in linked lists.

Criterion MP2. When Bob explores simple text search, COFALE presents a hyperlink encouraging Bob to examine the related concept “linked lists”. Similarly, while exploring this concept, Bob could return to the recursion hyperspace by using one of the hyperlinks presented in “Related Topics” and “Learning History” (the latter contains the hyperlinks of Bob’s recently visited content pages that are generated by COFALE). The two menus (Figure 1) also help Bob navigate intelligently to avoid getting lost in the learning hyperspace.


Recursion
Learning situations
Arithmetic expressions

Contents:
- Recursive definition
- Recursive evaluation
- Recursive evaluation process
- Java implementation
- Java test class

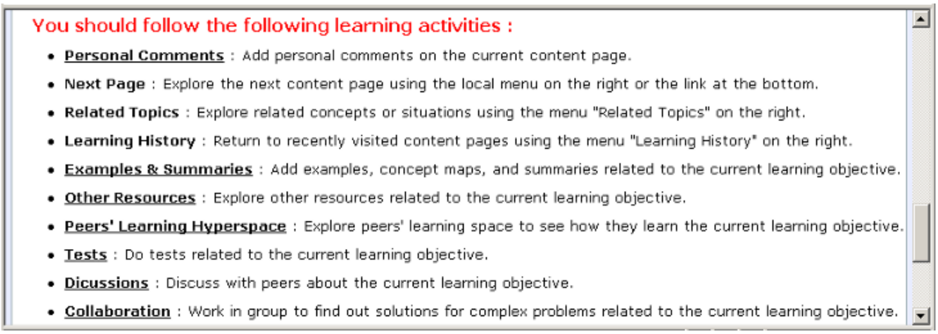
Simple arithmetic expressions: A simple arithmetic expression composes of integer numbers and simple operators +, -, *, and / (integer division), for example, E = (2 + 3) * 4 - 3 * (3 - 1). We often use a binary tree to represent an arithmetic expression, as the following figure.

A recursive representation of the expression (2 + 3) * 4 - 3 * (3 - 1)

Figure 1. A Part of Bob's Learning Hyperspace in COFALE

To satisfy criterion MP2, the course designer has defined, for every discrete piece of learning content (page), the other pages related to that one; e.g., simple text search related to arithmetic expressions, linked lists related to simple text search. On the basis of those associations, COFALE automatically generates the hyperlinks in "Related Topics" (Figure 1).

Criterion MM2. At the bottom of each content page, COFALE presents Bob with learning activities to guide and encourage him in the exploration of the learning hyperspace. For instance, after exploring arithmetic expressions, Bob is led to multiple activities in different contexts to look further into the concept of recursion (Figure 2).


You should follow the following learning activities :

- **Personal Comments** : Add personal comments on the current content page.
- **Next Page** : Explore the next content page using the local menu on the right or the link at the bottom.
- **Related Topics** : Explore related concepts or situations using the menu "Related Topics" on the right.
- **Learning History** : Return to recently visited content pages using the menu "Learning History" on the right.
- **Examples & Summaries** : Add examples, concept maps, and summaries related to the current learning objective.
- **Other Resources** : Explore other resources related to the current learning objective.
- **Peers' Learning Hyperspace** : Explore peers' learning space to see how they learn the current learning objective.
- **Tests** : Do tests related to the current learning objective.
- **Dicussions** : Discuss with peers about the current learning objective.
- **Collaboration** : Work in group to find out solutions for complex problems related to the current learning objective.

Figure 2. Learning Activities Proposed to Bob by COFALE

To satisfy criterion MM2, the course designer has defined, for each content page (e.g. "Java test class", the last item of arithmetic expressions in Figure 1), the learning activities related to that content page (e.g. the 10 activities shown in Figure 2). To help the course designer in this work, COFALE supports a set of predefined learning activities.

Criterion MP3. To satisfy this criterion, COFALE engages Bob in four learning activities: (1) add comments on the learning content proposed by the course designer, e.g. reformulate the main points of the definition of recursion (Figure 2: Personal Comments); (2) add his own examples, e.g. a recursive phenomenon in his life (Figure 2: Examples & Summaries); (3) explore external resources, e.g. the online Java tutorial [14] in which the author illustrates a great number of recursive examples (Figure 2: Other Resources); and (4) explore peers' learning spaces, e.g. log into the learning hyperspace of an "expert" learner to see and give feedback on her own recursive examples (Figure 2: Peers' Learning Hyperspace).

To support the exploration of external resources, the course designer has needed to search the Internet and introduce the chosen links (e.g. the Java tutorial [14]). The other three activities are supported by COFALE without explicit intervention of the course designer.

Criterion MP4. To satisfy this criterion, COFALE engages Bob to produce summaries of the points of view of other sources and peers (Figure 2: Examples & Summaries), e.g. produce a table stating his conceptions about recursion, together with possible misconceptions. COFALE supports this activity without intervention of the course designer.

Criterion MM3. To satisfy this criterion, COFALE encourages Bob to work with others (Figure 2: Discussions, Collaboration), sometimes with the participation of the tutor, by using multiple communication tools supported by ATutor – thus also by COFALE – such as e-mail, forums, chat rooms. COFALE also incites Bob to use Q&A websites to ask experts questions about recursion. The platform supports multiple communication tools, but to engage learners to use them, the course designer has created a forum and invited Bob and his peers to confront and compare their recursive solutions of a complex problem (listing all files and sub-directories of a given directory in a tree-structured file system).

Criterion MP5. To satisfy this criterion, COFALE presents a dropdown list of general and domain-specific questions that Bob could use to elicit peers' point of view. For instance, when Bob sees a recursive example or solution proposed by a peer, Bob can select the question "What was your source of information?" from the list to ask the peer to justify the solution.

The course designer is asked to prepare a list of general questions and a list of domain-specific questions. COFALE supports a list of predefined general questions proposed by researchers in pedagogy (e.g. [19]).

In addition to the previous situations, at any time Bob may review his learning behavior, i.e. navigation history, e.g. the content pages he has viewed and duration of each view (supported by ATutor). He can also see the tutor's feedback on his learning behavior with respect to CF (supported by COFALE); the tutor could explicitly ask him, for instance, to examine multiple learning situations in order to try and transfer the recursion concept in diverse contexts.

3.2 Learning with Support for Adaptability

We shall now assume that two other learners (Ted and Alice, both at the "expert" level) are active in the course: They are well versed in the use of COFALE and they have reached the analysis-synthesis model of the recursion concept. We now describe how the course designer uses COFALE to adapt the learning contents, pedagogical devices, and communication support to the specific needs of Bob, Ted, and Alice.

Learning contents. COFALE presents each learner with different content pages, e.g. simpler situations and examples for Bob than for Ted and Alice. To allow COFALE to perform this adaptation, the course designer has first decomposed the learning content into short content pages; then, the appropriate content pages are selected for each kind of learner, according to their mental models regarding recursion. This, of course, is a step in which the teacher's understanding of the various mental models among learners is essential.

Pedagogical devices. Because Bob is a "novice" and Ted and Alice are "experts", we must guide and encourage Bob much more than Ted and Alice in the learning process. For instance, COFALE suggests 10 activities (Figure 2) to Bob but only 5 "advanced" tasks to Ted and Alice (Figure 2: Personal Comments, Examples & Summaries, Tests, Discussions, Collaboration). To make this possible, the course designer is asked to define, for each content page, the appropriate learning activities for each type of learner.

Communication support. While learning with COFALE, learners can use a tool to search for peers who could help them overcome difficulties about acquiring the concept of recursion; COFALE may, for instance, suggest Ted and Alice to Bob so that he can ask them questions about simple problems; COFALE may suggest Ted to Alice so that they can exchange ideas about advanced recursive techniques. The course designer needs to define, for each kind of learner (according to the assumed mental model), the appropriate peers (e.g. learners with more-advanced mental models for learners with less-advanced ones).

At the beginning of the course, the course designer sets a default model for every new learner (e.g. the loop model in the case of recursion). During the learning process, three kinds of evaluations of mental models may be performed: (1) self-evaluation (e.g. after exploring situations and doing tests, Bob could identify that he possesses the analytic model); (2) evaluation by the tutor (e.g. after evaluating Bob's tests and learning behavior, the tutor could diagnose that Bob has reached the syntactic model); and (3) evaluation by COFALE (e.g. on the basis of Bob's test results provided by the tutor, COFALE could detect that Bob possesses the syntactic model). At certain times, e.g. after a test, learners may be asked to update the information about their mental model and choose the kind of evaluation they prefer; Bob, for instance, decides to always rely on his own evaluation. COFALE will immediately adapt the learning contents, pedagogical devices, and communication support to the new mental model. See [3, 12, 18] for more details about the various techniques for the three kinds of evaluation.

4. Discussion

The conclusion we draw from section 3 is that the use of COFALE we described earlier satisfies all the criteria for CF presented in section 1 in order to provide learners with *appropriate* learning conditions so that learners *actively construct* their own knowledge through their own learning activities. Note that the course designer's workload for making a course available in COFALE is not very high (about 8 person-hours for the course on recursion), because COFALE supports many learning activities without intervention of the course designer. In what follows, we discuss several issues on related work and on the implementation of COFALE.

We have analyzed several existing learning systems with respect to the criteria for CF [6] and adaptation techniques [5]; because of limited space, we show here only the result of our analysis. Firstly, we have looked into three learning systems that explicitly claim to support constructivism: KBS [12], Moodle [7], and SimQuest [10]. According to the available information and based on the set of criteria for CF, we have been able to construct Table 2. From this table, we may conclude that COFALE fills a number of shortcomings of those systems, especially in the area of pedagogical devices. Secondly, we have examined adaptation support in the following systems: AHA [9], ELM-ART [18], KBS [12], and PHelpS [11]. Table 3 shows that adaptation support in COFALE is comparable to that present in those systems.

Table 2. Conformity of Existing Learning Systems and COFALE with CF

Existing Learning Systems	Operational Criteria for CF							
	MM1	**MP1**	**MM2**	**MP2**	**MP3**	**MP4**	**MM3**	**MP5**
KBS	X	X	X	X				
Moodle	X		X		X		X	X
SimQuest	X	X	X	X				
COFALE	X	X	X	X	X	X	X	X

Table 3. Adaptation Support in Existing Learning Systems and COFALE

Existing Learning Systems	Presentation of Learning Contents	Presentation of Pedagogical Devices	Communication Support	Problem-Solving Support
AHA	X			
ELM-ART	X	X		X
KBS	X	X		
PHelpS			X	
COFALE	X	X	X	

For the implementation of COFALE, we have modified several components of ATutor and added the pedagogical devices and the learner model manager. Our contribution to ATutor is about 20 percent of the source code (6 person-months of programming work). We have selected ATutor among many open-source LCMSs because it makes it easy to add pedagogical devices exhibiting the desired characteristics for CF and to create and manage fine-grained sharable content objects that are compliant with the IMS/SCORM standard [15]. This characteristic is useful both for the design of goal-based learning and for the personalization of learning contents [15].

Conclusion

We have shown how to design and use adaptive learning environments supporting cognitive flexibility (CF), one important facet of constructivism. Our approach, based on operational criteria, has shown that the domain-independent platform COFALE truly supports learning conditions leading to CF. COFALE also supports several adaptation techniques borrowed from other adaptive learning systems. We believe that our approach could also be used for exploiting other facets of constructivism (e.g. problem solving) and other adaptation techniques (e.g. problem-solving support), leading to more completely constructivist and adaptive learning environments.

In the future, we plan to look into two additional issues: (1) how to help teachers evaluate their teaching behavior with respect to CF and (2) how to integrate domain-specific tools into COFALE to support problem solving by learners. In addition, we shall try and evaluate how effectively students learn recursion with the help of COFALE.

References

[1] Anderson, J.R., Pirolli, P., & Farrell, R., “Learning to Program Recursive Functions”, In M. Chi, R. Glaser, and M. Farr (Eds), *The Nature of Expertise*, Hillsdale, NJ: Erlbaum, 1988, pp. 153–184.

[2] ATRC group, *ATutor Platform*, Retrieved November 8, 2004 from: http://www.atutor.ca.

[3] Bhuiyan, S., Greer, J., & McCalla, G., “Supporting the Learning of Recursive Problem Solving”, *Interactive Learning Environments, 4(2)*, 1994, pp. 115–139.

[4] Bruner, J.S., *Going Beyond the Information Given*, New York: Norton, 1973.

[5] Brusilovsky, P., “Adaptive and Intelligent Technologies for Web-based Education”, In C. Rollinger and C. Peylo, *Special Issue on Intelligent Systems and Teleteaching, Künstliche Intelligenz, 4*, 1999, pp. 19–25.

[6] Chieu, V.M., Milgrom, E., & Frenay, M., “Constructivist Learning: Operational Criteria for Cognitive Flexibility”, *The 4th IEEE International Conference on Advanced Learning Technology*, 2004, pp. 221–225.

[7] Dougiamas, M., *Moodle Platform*, Retrieved November 8, 2004 from: http://moodle.org.

[8] Driscoll, M.P., *Psychology of Learning for Instruction*, Massachusetts: Allyn and Bacon, 2000.

[9] De Bra, P. & Calvi, L., "AHA! An Open Adaptive Hypermedia Architecture", *The New Review of Hypermedia and Multimedia, 4*, 1998, pp. 115–139.

[10] De Jong, T., van Joolingen, W., & van der Meij, J., *SimQuest Discovery Learning*, Retrieved November 8, 2004 from: http://www.simquest.nl.

[11] Greer, J., McCalla, G., Collins, J., Kumar, V., Meagher, P., & Vassileva, J., "Supporting Peer Help and Collaboration in Distributed Workplace Environments", *IJAIED, 9*, 1998, pp. 159–177.

[12] Henze, N. & Nejdl, W., “Adaptation in Open Corpus Hypermedia”. *IJAIED, 12*, 2001, pp. 325–350.

[13] Kinshuk, Looi, C.K., Sutinen, E., Sampson, D., Aedo, I., Uden, L., & Kähkönen, E., *Proceeding of the 4th IEEE International Conference on Advanced Learning Technologies*, IEEE Computer Society, 2004.

[14] Kjell, B., *Introduction to Computer Science Using Java*, Retrieved November 8, 2004 from: http://chortle.ccsu.edu/CS151/cs151java.html.

[15] Masie Center, *Making Sense of Learning Standards and Specifications*, Retrieved November 8, 2004 from: http://www.masie.com/standards/s3_2nd_edition.pdf.

[16] Santrock, J.W., *Educational Psychology*, NewYork: McGraw-Hill, 2001.

[17] Spiro, R.J. & Jehng, J.C., “Cognitive Flexibility and Hypertext: Theory and Technology for the Nonlinear and Multidimensional Traversal of Complex Subject Matter”, In D. Nix and R.J. Spiro: *Cognition, Education and Multimedia*, Hillsdale, NJ: Erlbaum, 1990.

[18] Weber, G. & Brusilovsky, P., “ ELM-ART: an Adaptive Versatile System for Web-based Instruction”, *IJAIED, 12*, 2001, pp. 351–384.

[19] Wright, W.A., *Teaching Improvement Practices. Successful Strategies for Higher Education*, Bolton: Anker Publishing Company, 1995.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

The Effect of Explaining on Learning: a Case Study with a Data Normalization Tutor

Antonija MITROVIC
Intelligent Computer Tutoring Group
Department of Computer Science and Software Engineering
University of Canterbury, New Zealand

Abstract: Several studies have shown that explaining actions increases students' knowledge. In this paper, we discuss how NORMIT supports self-explanation. NORMIT is a constraint-based tutor that teaches data normalization. We present the system first, and then discuss how it supports self-explanation. We hypothesized the self-explanation support in NORMIT would result in increased problem solving skills and better conceptual knowledge. An evaluation study of the system was performed, the results of which confirmed our hypothesis. Students who self-explained learnt constraints significantly faster, and acquired more domain knowledge.

1. Introduction

Although Intelligent Tutoring Systems (ITS) result in significant learning gains [9,11,12,13, 19], some empirical studies indicate that even in the most effective systems, some students acquire shallow knowledge. Examples include situations when the student can guess the correct answer, instead of using the domain theory to derive the solution. Aleven et al. [1] illustrate situations when students guess the sizes of angles based on their appearance. As the result, students have difficulties in transferring knowledge to novel situations, even though they obtain passing grades on tests.

The goal of ITSs is to enable students to acquire deep, robust knowledge, which they can use to solve different kinds of problems, and to develop effective meta-cognitive skills. Psychological studies [6,7] show that self-explanation is one of the most effective learning strategies. In self-explanation, the student solves a problem (or explains a solved problem) by specifying why a particular action is needed, how it contributes toward the solution of the problem, and what basic principles of the domain were used to perform the action.

This paper presents the support for self-explanation in NORMIT, a data normalization tutor. Section 2 reviews related work. Section 3 overviews the learning task, while the support for self-explanation is discussed in Section 4. The results of an evaluation study of NORMIT are presented in Section 5. The conclusions and avenues for future research are given in the final section.

2. Related Work

Metacognition includes processes involved with awareness of, reasoning and reflecting about, and controlling one's cognitive skills and processes. Metacognitive skills can be taught [5], and result in improved problem solving and better learning [1,8,18]. Of all metacognitive skills, self-explanation (SE) has attracted most interest within the ITS community. By explaining to themselves, students integrate new knowledge with existing knowledge. Furthermore, psychological studies show that self-explanation helps students to correct their misconceptions [7]. Although many students do not spontaneously self-explain, most will do so when prompted [8] and can learn to do it effectively [5].

SE-Coach [8] is a physics tutor that supports students while they study solved examples. The authors claim that self-explanation is better supported this way, than asking for explanation while solving problems, as the latter may put too big a burden on the student. In this system, students are prompted to explain a given solution for a problem. Different parts of the solution are covered with boxes, which disappear when the mouse is positioned over them. This masking mechanism allows the system to track how much time the student spends on each part of the solution. The system controls the process by modelling the self-explanation skills using a Bayesian network. If there is evidence that the student has not self-explained a particular part of the example, the system will require the student to specify why a certain step is correct and why it is useful for solving the current problem. Empirical studies performed show that this structured support is beneficial in early learning stages.

On the other hand, Aleven and Koedinger [1] explore how students explain their own solutions. In the PACT Geometry tutor, as students solve problems, they specify the reason for each action taken, by selecting a relevant theorem or a definition from a glossary. The performed evaluation study shows that such explanations improve students problem-solving and self-explanation skills and also result in transferable knowledge. In Geometry Explanation Tutor [2], students explain in natural language, and the system evaluates their explanations and provides feedback. The system contains a hierarchy of 149 explanation categories [3], which is a library of common explanations, including incorrect/incomplete ones. The system matches the student's explanation to those in the library, and generates feedback, which helps the student to improve his/her explanation.

In a recent project [21], we looked at the effect of self-explanation in KERMIT, a database design tutor [19,20]. In contrast to the previous two systems, KERMIT teaches an open-ended task. In geometry and physics, domain knowledge is clearly defined, and it is possible to offer a glossary of terms and definitions to the student. Conceptual database design is a very different domain. As in other design tasks, there is no algorithm to use to derive the final solution. In KERMIT, we ask the student to self-explain only in the case their solution is erroneous. The system decides on which errors to initiate a self-explanation dialogue, and asks a series of question until the student gives the correct answer. The student may interrupt the dialogue at any time, and correct the solution. We have performed an experiment, the results of which show that students who self-explain acquire more conceptual knowledge than their peers [22].

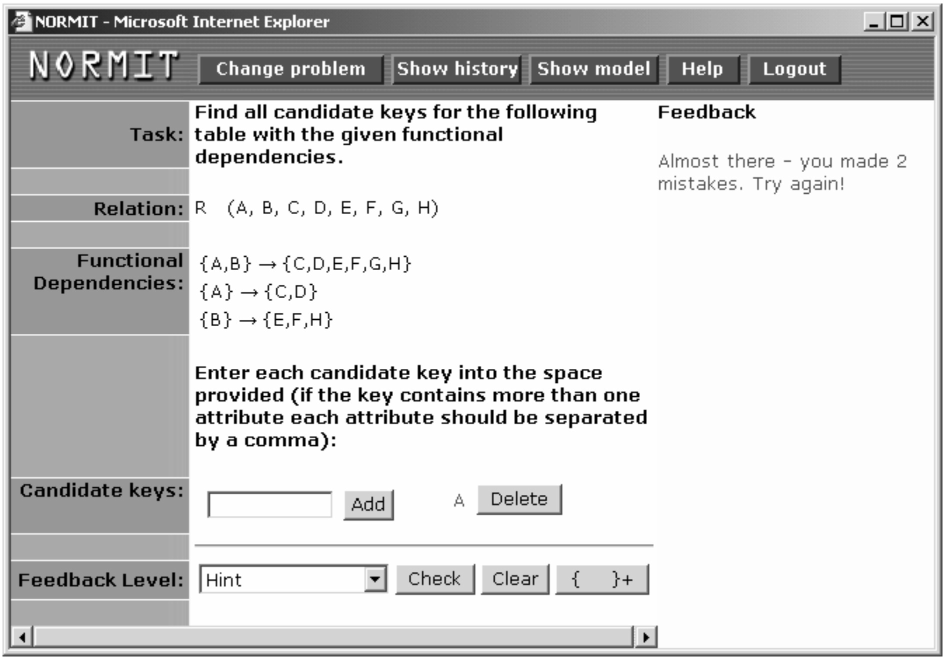
3. Learning Data Normalization in NORMIT

Database normalization is the process of refining a relational database schema in order to ensure that all tables are of high quality [10]. Normalization is usually taught in introductory database courses in a series of lectures that define all the necessary concepts, and later practised on paper by looking at specific databases and applying the definitions.

Like other constraint-based tutors [13,14,19], NORMIT is a problem-solving environment, which complements traditional classroom instruction. The emphasis is therefore on problem solving, not on providing information. Database normalization is a procedural task: the student goes through a number of steps to analyze the quality of a database. NORMIT requires the student to determine candidate keys (Figure 1), the closure of a set of attributes and prime attributes, simplify functional dependencies, determine normal forms, and, if necessary, decompose the table. The sequence is fixed: the student will only see a Web page corresponding to the current task. The student may submit a solution or request a new problem at any time. He/she may also review the history of the session, or examine the student model.

When the student submits the solution, the system analyses it and offers feedback. The first submission receives only a general feedback, specifying whether the solution is correct or not (as in Figure 1). If there are errors in the solution, the incorrect parts of the solution are shown in red. In Figure 1, for example, the student has specified A as the key of the given relation, which is incorrect. On the second submission, NORMIT provides a general description of the error, specifying what general domain principles have been violated. On the third submission, the system provides a more detailed message, by providing a hint as to how the student should change the solution. The student can also get a hint for every error. The correct solution is only available on request.

NORMIT is a Web-enabled tutor with a centralized architecture. As NORMIT is a constraint-based tutor [13,17], the domain model is represented as a set of 81 problem-independent constraints. For details of the system's architecture and implementation, please see [15].



Task: Find all candidate keys for the following table with the given functional dependencies.

Relation: R (A, B, C, D, E, F, G, H)

Functional Dependencies: {A,B} → {C,D,E,F,G,H}
{A} → {C,D}
{B} → {E,F,H}

Enter each candidate key into the space provided (if the key contains more than one attribute each attribute should be separated by a comma):

Candidate keys: A

Feedback Level: Hint

Feedback

Almost there - you made 2 mistakes. Try again!

Fig. 1. A screenshot from NORMIT

4. Supporting Self-Explanation

NORMIT is a problem-solving environment, and therefore we ask students to self-explain while they solve problems. In contrast to other ITSs that support self-explanation, we do not expect students to self-explain every problem-solving step. Instead, NORMIT will require an explanation for each action that is performed for the first time. For the subsequent actions of the same type, explanation is required only if the action is performed incorrectly. We believe that this strategy will reduce the burden on more able students (by not asking them to provide the same explanation every time an action is performed correctly), and also that the system would provide enough situations for students to develop and improve their self-explanation skills.

Similar to the PACT Geometry Tutor and SE-Coach, NORMIT supports self-explanation by prompting the student to explain by selecting one of the offered options. In Figure 1, the student specified A as the candidate key incorrectly. NORMIT then asks the following question (the order in which the options are given is random, to minimize guessing):

This set of attributes is a candidate key because:

- *It is a minimal set of attributes*
- *Every value is unique*
- *It is a minimal set of attributes that determine all attributes in the table*
- *It determines the values of all other attributes*
- *All attributes are keys*
- *Its closure contains all attributes of the table*

The candidate answers to choose from are not strict definitions from the textbook, and the student needs to reason about them to select the correct one for the particular state of the problem. For this reason, we believe that the support for self-explanation in NORMIT (i.e. explanation selection) is adequate support. In this way, self-explanation is not reduced to recognition, but truly requires the student to re-examine his/her domain knowledge in order to answer the question. Therefore, this kind of self-explanation support requires recall and is comparable to generating explanations. Furthermore, this kind of self-explanation support is easier to implement in comparison to explaining in a natural language. Although it may seem that explaining in a natural language would give better results than selecting from pre-specified options, Aleven, Koedinger and Popescu [4] show that this is not necessarily the case: in their study there was no significant difference between students who explained by selecting from menus, and students who explained in English.

If the student's explanation is incorrect, he/she will be given another question, asking to define the underlying domain concept (i.e. candidate keys). For the same situation, the student will get the following question after giving an incorrect reason for specifying attribute A as the candidate key:

A candidate key is:

- *an attribute with unique values*
- *an attribute or a set of attributes that determines the values of all other attributes*
- *a minimal set of attributes that determine all other attributes in the table*
- *a set of attributes the closure of which contains all attributes of the table*
- *a minimal superkey*
- *a superkey*
- *a key other than the primary key*
- *A candidate key is an attribute or a set of attributes that determine all other attributes in the table and is minimal. The second condition means that it is not possible to remove any attributes from the set, and still have the remaining attributes to determine the other attributes in the table.*

In contrast to the first question, which was problem-specific, the second question is general. If the student selects the correct option, he/she will resume with problem solving. In the opposite case, NORMIT will provide the correct definition of the concept.

In addition to the model of the student's knowledge, NORMIT also models the student's self-explanation skills. For each constraint, the student model contains information about the student's explanations related to that constraint. The student model also stores the history of student's explanation of each domain concept.

5. Experiment

We performed an evaluation study with the students enrolled in an introductory database course at the University of Canterbury. Our hypothesis was that self-explanation would have positive effects on both procedural knowledge (i.e. problem solving skills) and conceptual knowledge. Prior to the experiment, the students had four lectures and one tutorial on data normalization. The system was demonstrated in a lecture on October 5, 2004 (during the last week of the course), and was open to the students a day later. The students in the control group used the basic version of the system, while the experimental group used NORMIT-SE, the version of the system that supports self-explanation. The participation was voluntary, and 61 out of 124 students enrolled in the course used the system. The students were free to use NORMIT when and for how long they wanted.

The pre-test (with the maximum mark of 4) was administered on-line at the beginning of the first session. We developed two tests, each having four multichoice questions. The first two questions required students to identify the correct solution for a given problem, while for the other two students needed to identify the correct definition of a given concept. These two tests were randomly used as the pre-test. The post-test was administered as a part of the final examination on October 29, 2004.

Table 1. Mean system interaction details (standard deviations given in parentheses)

	NORMIT	**NORMIT-SE**
Students	27	22
Sessions	2.9 (1.95)	2.4 (1.7)
Time spent (min.)	231 (202)	188 (167)
Attempted problems	**16.7 (11.2)**	**11.9 (10.4)**
Completed problems (%)	81.9 (22.5)	80.4 (16.2)
Pre-test (%)	55.6 (26.2)	64.77 (26.3)
Post-test (%)	51.3 (15.4)	53.61 (22.3)

We collected data about each session, including the type and timing of each action performed by the student, and the feedback obtained from NORMIT. Twelve students have logged on to the system for a very short time, and have solved no problems, and we excluded their logs from analyses. Table 1 reports some statistics about the remaining students. The average mark on the pre-test for all students was 59.7% (sd = 26.4). The groups are comparable, as there is no significant difference on the pre-test.

There was no significant difference between the two groups on the number of sessions or the total time spent with the system. The number of attempted problems ranged from 1 to 49 (the total number of problems in the system is 50). The difference between the mean number of attempted problems for the two groups is significant (p=0.067). We believe this is due to more time needed for self-explanation for the experimental group students. Both groups of students were equally successful at solving problems, as there was no significant difference on the percentage of solved problems.

As explained earlier, the post-test was administered as a part of the final examination for the course. We decided to measure performance this way because the study was not controlled, and this was the only way to ensure that each participant sits the post-test. However, this decision also dictated the kinds of questions appearing in the post-test. As the consequence, our pre- and post-tests are not directly comparable. The post-test was longer, with a maximum of 29 marks. Therefore we cannot compare the students' performance before and after the study.

There was no significant difference between the post-test results of the two groups. However, it is important to note that 60% of the control group students and 73% of the experimental group students logged on to NORMIT for the first time just a day or two before the post-test. Furthermore, the students on average spent only 3-4 hours working with the system. Therefore, it is not reasonable to expect significant difference after such short interaction times.

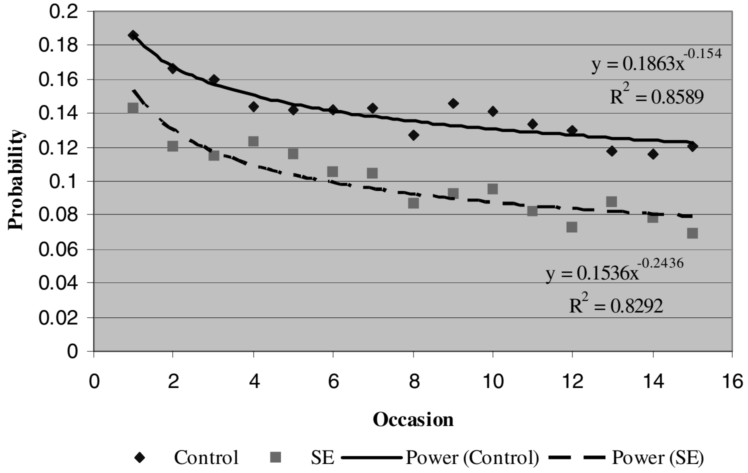
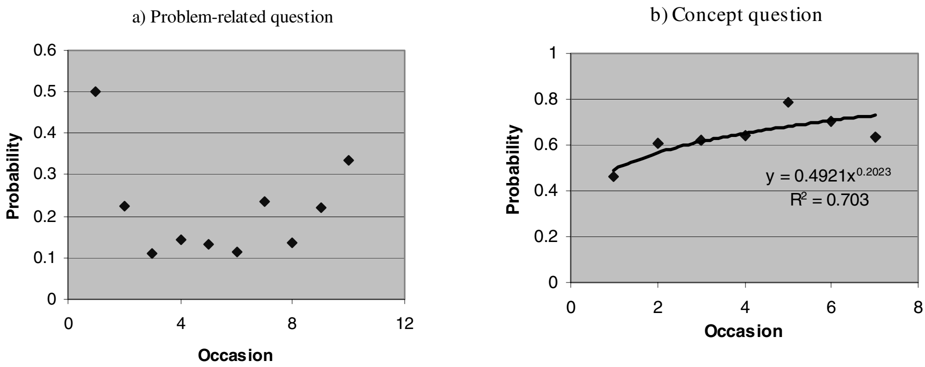


Fig. 2. Learning constraints

Figure 2 shows how students learnt constraints. We looked at the proportion of violated constraints following the n^{th} occasion when a constraint was relevant, averaged across all students and all constraints. The R^2 fits to the power curves are good for both groups, showing that all students learnt constraints by using the system. The learning curve for the experimental group shows that these students are less likely to violate constraints and learn constraints faster than their peers. The learning rate of the experimental group (.24) is higher than the learning rate of the control group (.15). We have also analysed individual learning curves, for each participant in the study. The learning rates of students in the experimental group are significantly higher than those of the control group students (p=0.014). This finding confirms our hypothesis that self-explanation has a positive effective on students' domain knowledge.

We also analysed the data about students' self-explanations. There were 713 situations where students were asked to self-explain. On average, a student was asked 32.4 problem-oriented SE questions (i.e. the first question asked when a student makes a mistake), and 23.2 concept-oriented SE questions, and correct explanations were given in 31.9% and 56.7% of the cases respectively. Figure 3.a shows the probability of giving a correct answer to the problem-related SE question averaged over all occasions and all participants. As can be seen, this probability varies over occasions, but always stays quite low. Therefore, students find it hard to give reasons for their actions in the context of the current problem. Some concepts are much more difficult for students to learn than others. For example, out of the total of 132 situations when students who were asked to explain why a set of attributes is a candidate key, the correct answer was given in only 23 cases. Figure 3.b shows the same probability for the question asking to define a domain concept (conceptual question). As the figure illustrates, the students were much better at giving definitions of domain concepts. In the case of candidate keys, although students were pretty bad in justifying their choice of candidate key in a particular situation (when the correct answer was given in 17.4% of the cases), when asked to define a candidate key, they were correct in 45% of the cases. Figure 3.b shows a regular increase of the probability of correct explanation, showing that the students did improve their conceptual knowledge through explaining their actions.


a) Problem-related question

b) Concept question

Fig. 3. Student's performance on self-explanation

6. Conclusions

Self-explanation is known to be an effective learning strategy. Since ITSs aim to support good learning practices, it is not surprising that researches have started providing support for self-explanation. In this paper, we present NORMIT-SE, a data normalization tutor, and describe how it supports self-explanation. NORMIT-SE is a problem-solving environment, and students are asked to explain their actions while solving problems. The student must explain every action that is performed for the first time. However, we do not require the student to explain every action, as that would put too much of a burden on the student and reduce motivation. NORMIT-SE requires explanations in cases of erroneous actions. The student is asked to specify the reason for the action, and, if the reason is incorrect, to define the domain concept that is related to the current task. If the student is not able to identify the correct definition from a menu, the system provides the definition of the concept.

We performed a pilot study of the system in a real course in 2002 [16]. In 2003 we performed an evaluation study, but did not have enough participants to draw any conclusions. This paper presented a study performed in 2004, which had more participants than the previous two. The results of the study support our hypothesis: students who self-explained learned constraints significantly faster than their peers who were not asked to self-explain. There was no significant difference between the two conditions on the post-test performance, and we believe that is due to the short times the participants spent interacting with the system. Furthermore, the analysis of the self-explanation behaviour shows that students find problem-specific question (i.e. explaining their action in the context of the current problem state) more difficult than defining the underlying domain concepts. The students' conceptual knowledge improved regularly during their interaction with NORMIT-SE.

There are two main avenues for future work. At the moment, the student model in NORMIT contains a lot of information about the student's self-explanation skills that is not used. We plan to use this information to identify domain concepts for which the student needs more instruction. Furthermore, the self-explanation support itself may be made adaptive, so that different support would be offered to students who are poor self-explainers in contrast to students who are good at it.

Acknowledgements: We thank Li Chen and Melinda Marshall for implementing NORMIT's interface.

References

1. Aleven, V., Koedinger, K., Cross, K. Tutoring Answer Explanation Fosters Learning with Understanding. In *Proc. Int. Conf. Artificial Intelligence and Education*, 1999, pp. 199-206.
2. Aleven, V., Popescu, O., Koedinger, K. Towards Tutorial Dialogue to Support Self-Explanation: Adding Natural Language Understanding to a Cognitive Tutor. Int. J. Artificial Intelligence in Education, vol. 12, 2001, 246-255.
3. Aleven, V., Popescu, O., Koedinger, K. Pilot-Testing a Tutorial Dialogue System that Supports Self-Explanation. In *Proc. Int. Conf. Intelligent Tutoring Systems*, Biarritz, France, 2002, pp. 344-354.
4. Aleven, V., Popescu, O., Koedinger, K. A Tutorial Dialogue System to Supports Self-Explanation: Evaluation and Open Questions. In U. Hoppe, F. Verdejo and J. Kay (eds) *Proc. Int. Conf. Artificial Intelligence in Education,* Sydney, 2003, pp. 39-46.
5. Bielaczyc, K., Pirolli, P., Brown, A. Training in Self-Explanation and Self-Regulation Strategies: Investigating the Effects of Knowledge Acquisition Activities on Problem-solving. Cognition and Instruction, vol. 13, no. 2, 1993, 221-252.
6. Chi, M. Self-explaining Expository Texts: The dual processes of generating inferences and repairing mental models. Advances in Instructional Psychology, 2000, 161-238.
7. Chi, M. Bassok, M., Lewis, W., Reimann, P., Glaser, R. Self-Explanations: How Students Study and Use Examples in Learning to Solve Problems. Cognitive Science, vol. 13, 1989, 145-182.
8. Conati, C., VanLehn, K. Toward Computer-Based Support of Meta-Cognitive Skills: a Computational Framework to Coach Self-Explanation. Int. J. Artificial Intelligence in Education, vol. 11, 2000, 389-415.
9. Corbett, A., Trask, H., Scarpinatto, K., Handley, W. A formative evaluation of the PACT Algebra II Tutor : support for simple hierarchical reasoning. In *Proc. Int. Conf. Intelligent Tutoring Systems*, San Antonio, 1998, pp. 374-383.
10. Elmasri, R., Navathe, S. B. *Fundamentals of database systems*. Benjamin/Cummings, 2003.
11. Gertner, A.S, VanLehn, K. ANDES: A Coached Problem-Solving Environment for Physics. In G. Gauthier, C. Frasson, and K. VanLehn, (eds.), *Proc. Int. Conf. ITS*, Montreal, 2000, pp. 133-142.
12. Grasser, A., Wiemer-Hastings, P., Kreuz, R. AUTOTUTOR: A Simulation of a Human Tutor. Journal of Cognitive Systems Research, vol. 1, no. 1, 1999, 35-51.
13. Mitrovic, A., Ohlsson, S. Evaluation of a constraint-based tutor for a database language. Int. J. Artificial Intelligence in Education, vol. 10, no. 3-4, 1999, 238-256.
14. Mitrovic, A., Suraweera, P., Martin, B, Weerasinghe, A. DB-suite: Experiences with Three Intelligent, Web-based Database Tutors. Interactive Learning Research, vol. 15, no. 4, 409-432.
15. Mitrovic, A. NORMIT, a Web-enabled tutor for database normalization. In Proc. Int. Conf. Computers in Education, Auckland, New Zealand, 2002, pp. 1276-1280.
16. Mitrovic, A. Supporting Self-Explanation in a Data Normalization Tutor. In: V. Aleven, U. Hoppe, J. Kay, R. Mizoguchi, H. Pain, F. Verdejo, K. Yacef (eds) Supplementary proceedings, AIED 2003, 2003, pp. 565-577.
17. Ohlsson, S. Constraint-based Student Modeling. In Student Modeling: the Key to Individualized Knowledge-based Instruction. 1994, 167-189.
18. Schworm, S., Renkl, A. Learning by solved example problems: Instructional explanations reduce self-explanation activity. Proc. 24th Cognitive Science Conf., 2002, pp. 816-821.
19. Suraweera, P., Mitrovic, A. An Intelligent Tutoring System for Entity Relationship Modelling. Int. J. Artificial Intelligent in Education, vol. 14, no. 3-4, 2004, 375-417.
20. Suraweera, P., Mitrovic, A. KERMIT: a Constraint-based Tutor for Database Modeling. In *Proc .Int. Conf. Intelligent Tutoring Systems,* Biarritz, France, 2002, pp. 377-387.
21. Weerasinghe, A., Mitrovic, A. Enhancing learning through self-explanation. *Proc. Int. Conf. Computers in Education*, Auckland, New Zealand, 2002, pp. 244-248.
22. Weerasinghe, A., Mitrovic, A. Supporting Self-Explanation in an Open-ended Domain. In: M. Gh. Negoita, R. J. Howlett and L. C. Jain (eds) Proc. 8th Int. Conf. Knowledge-Based Intelligent Information and Engineering Systems KES 2004, Berlin, Springer LNAI 3213, 2004, pp. 306-313.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Formation of Learning Groups by using Learner Profiles and Context Information

Martin Muehlenbrock
German Research Center for Artificial Intelligence DFKI
66123 Saarbruecken, Germany, Martin.Muehlenbrock@dfki.de

Abstract. An important but often neglected aspect in Computer Supported Collaborative Learning is the intelligent formation of learning groups. Until recently, support for group formation was mostly based on learner profile information. However, the perspective of ubiquitous computing and ambient intelligence allows for taking a broader view on group formation, extending the range of features to include learner context information such as sensor-derived activity and availability. A probabilistic approach has been developed that automatically learns individual characteristics and indicates relevant situations, and which has been tested in a set of experiments.

1. Introduction

An important but often neglected aspect in Computer-Supported Collaborative Learning is the formation of learning groups. Most CSCL systems focus on mediating and supporting collaborative learning while the activity is going on, or after the activity has ended, by proving system functionality ranging from mirroring to guiding [6]. Moreover, if support could also be given prior to the actual collaborative learning activity by suggesting appropriate group arrangements, many problems might be solved even before they arise, and beneficial group processes might be boosted.

Until recently, most support for group formation was based on learner profile information such as gender, class, etc., including more sophisticated information such as the complementarity or overlapping of knowledge and competencies. Such an approach will be described in the following section. In addition, the perspective of ubiquitous computing and ambient intelligence allows for a wider perspective on group formation, broadening the range of addressed features to include learner context information such as location, time, and availability. This new perspective will be addressed in the third section.

1. Group Formation based on Learner Profiles

A general conceptual and formal framework for student model integration has been introduced in [3] under the notion of multiple student modelling, and has been extended in [10] for open distributed learning environments. The general premise is that individually assessed learner models can be used to support the configuration or parameterization of collaborative learning settings. These are prototypical cases:

- Given a number of students working on comparable problems in an open learning network, find pairs of students that could potentially benefit from cooperation in a joint

session. The selection can be based on such criteria as complementarity or competitiveness.
- Given a group of students, select or generate a problem that forms an adequate challenge for the group as a whole. The problem should not be solvable by one student's knowledge alone, but rather through the union of all the students' individual knowledge bases. In this case, the challenge for the group consists in knowledge exchange and integration.

Selection criteria for these prototypical cases can be formulated on the basis of general modelling primitives such as *knows(Student, Topic)* or *has_difficulty(Student, Topic)*, which can be inferred from different standard types of student models. A simple case of knowledge integration is exemplified by the rule

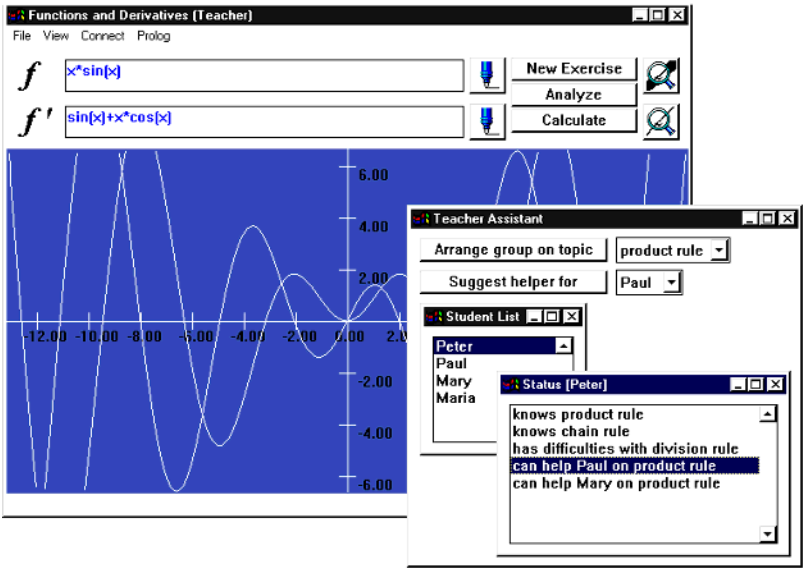
can_help(Student1, Student2, Topic) ←
knows(Student1, Topic) & has_difficulty(Student2, Topic).

Interestingly, there is a wide range of different support functions that can be implemented based on such a rule and further extensions:

- **Intelligently mediated peer help:** The individually assessed learner models are used to match pairs of learners that should maximally benefit from each other when working together. The prediction can be based on different criteria such as complementarity of skills/knowledge or competition.
- **Intelligently mediated expert tutoring:** Formally, this case can be considered as a specialization and simplification of matching peer learners, since only one of the models (the learner's) has to be dynamically assessed, whereas the tutors' profiles may be predefined.
- **Teacher/tutor support for supervising individual exercises:** Essentially a decision support function for the teacher. To achieve this it is sufficient to aggregate the individual learner models in a form that allows for filtering out specific features, e.g. frequent problems. The support mechanism can also actively inform the teacher if adequate.
- **Group formation around given problems:** This is a generalization of mediating peer help in that the number of group members is not restricted to two. Also the problem requirements must be analytically specified.
- **Selection of adequate problems for a given group:** A problem is e.g. selected or generated in such a way that it could serve as a challenge to the group as a whole but should still be feasible if the group were able to combine individual strengths.

This framework has been used in different learner grouping scenario. For instance, see figure 1 for a user interface that proposes peer helpers for a learning task in mathematics. In the context of group learning, the individual student models are accumulated and integrated to derive a model of group problem solving that initiates and supports remedial activities. The underlying distributed architecture of the intelligent subsystem must allow for combining elements from different individual student models, as has been described in [10].

Massive practical applications of group formation based on similar principles as described here have been reported by [7]. An ontology-based representation of group formation principles has been proposed by [5].



x*sin(x)

sin(x)+x*cos(x)

knows product rule
knows chain rule
has difficulties with division rule
can help Paul on product rule
can help Mary on product rule

Figure 1. User interface for the formation of learning groups including peer helper suggestion and topic selection.

3. Group Formation based on Learner Context

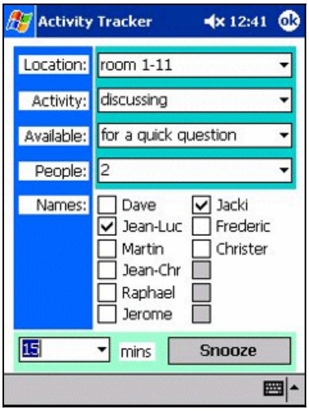
The concept of ubiquitous computing envisions a new computing era where computational and communication power is available in devices and objects of every size and purpose [12]. One of the biggest challenges in ubiquitous computing is the automatic detection of a user context [11]. A typical contextual variable of a user that is frequently addressed is location, driven by many advances in device and sensor technology. Further interesting context features of a user and in a user's environment include among others activity, availability, stress and emotional parameters as well as temperature, noise, weather, co-location of other people, and availability of devices, respectively. For learning group formation, these contextual features provide an additional source of learner information, which could help in improving the quality of the grouping.

Using a networked infrastructure of easily available sensors and context-processing components, an application has been developed for peer helper suggestion and opportunistic group formation based on contextual parameters such as location, activity, and availability [9]. These notions of location, activity, and availability have both been detected automatically based on sensor information and learnt automatically based on users' feedback to the system.

In order to detect a person's location, activity, and availability, different sensing techniques have been used in a prototypical application. All these sensors are already available in many environments or can be installed without much effort, such as

- **PDA location:** Determination of the location of user's PDA (personal digital assistant) by using a wireless network. Wireless LANs are becoming more and more widespread, and a location system can be obtained as a by-product of the wireless LAN by triangulating the radio signal [2]. Places are first identified by their radio characteristics such as signal strengths in a calibration phase. Afterwards a device can locate itself by measuring the current radio characteristic and comparing it with calibration data, resulting in a localization reliability of about 80% according to our experience [1].

- **PC usage:** Detection of users' keyboard and mouse activity on personal computers. Sensing the user activity level on a personal computer is an important and easy source of information. The PC usage is detected by a demon that runs on the PCs and monitors typing and mouse movements.
- **PDA ambient sound:** Detection of ambient sound in the PDAs' vicinities. Each PDA is equipped with a microphone that is used to record several sound samples in a minute. These sound samples are compared to a sample of those situations with the lowest sound level encountered so far, defining a reference point for the no-ambient-sound situation.
- **PDA user feedback:** Explicit feedback on some context variables provided by the users. A user interface has been developed for the PDA that prompts the user for information on his context in a regular fashion. This user information is used to label situations in order to create a set of training data for calibrating the context sensing system to individual characteristics. The user is asked to provide explicit feedback on a number of context variables. These include his location, the co-location with other people, which could be either people identified to the system or just the number of people present, activity and availability (see figure 2).



Location: room 1-11

Activity: discussing

Available: for a quick question

People: 2

Names:
□ Dave ☑ Jacki
☑ Jean-Luc □ Frederic
□ Martin □ Christer
□ Jean-Chr □
□ Raphael □
□ Jerome □

15 mins

Figure 2. PDA user interface for context feedback.

The various sensors send their information to a database residing on a server that can be accessed from both the wired and the wireless networks (see figure 3). The database contains static profile data as well as the dynamic event data. The static profile data may vary over time, e.g. if someone is allocated a new PC or changes office, but comparatively slowly compared to the event data. The profile data names the entities, i.e., people and devices, and places that are referred to by the dynamic event data. Furthermore the profile establishes links between devices and places and people. For example the profile indicates that particular computers, PDAs and phones are associated with a particular user and that a user has his office in a particular place. It also indicates the normal function of places so that our software can find out if a user is in a place that is someone's office or in a public space such as a meeting room or coffee area. The tables associated with the dynamic event data store information about events generated by the sensors as well as the events generated by higher-level components predicting activity and availability.

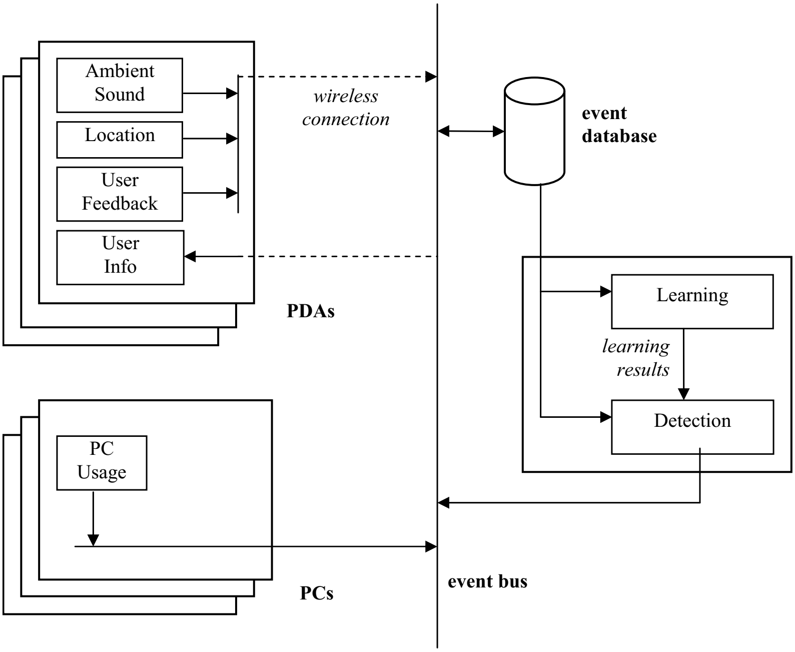


Figure 3. System architecture.

The context processing consists of combining information from different sources and deriving an estimation of the users' situation. Of particular interest for the application are the activities and availabilities of the users. The set of relevant activities is comprised of single-person activities like using a PC, using a PDA, and working on the desk, multi-person activities such as phoning, discussing, or being in a meeting, and intermediate activities like walking from one place to another, which result in a drastic change of context. These activities are assumed to have a major influence on the level of a person's availability. Relevant classes of availabilities that are considered to be useful are being available for a quick question, being available for a longer discussion, being available soon, or not being available at all. By using machine-learning methods the system is to find a connection between sensed information and situations as perceived by users, including also information on people's habits.

On the basis of labeled sensor data, probabilistic classifiers for relevant user activities and availabilities are learnt. As can be seen in figure 4, user activity is related to the PDA location, the PC usage, the ambient sound, the PDA co-location, and the time of day, whereas user availability is related to PDA location, activity, and time of day. A Bayesian approach is used to determine the activity with the maximum a posteriori probability. The simplifying assumption is made that all sensor values are conditionally independent (Naïve Bayesian classifier). The estimation of the prior probabilities for the Bayesian learning is based on the number of occurrences of each activity in the user feedback with and without the respective sensor value being detected as well as on the sum of probabilities of rooms in the user feedback where an activity was indicated. In order to get more reliable probability values, especially in the case of missing user feedback, a simple LaPlace smoothing has been used. Similarly, probabilistic classifiers for users' availabilities are derived.

The results of the learning of activity and availability notions are automatically included in a detection component, which is constantly monitoring the most recent events in the event database. For each user the detection component derives an up-to-date context description

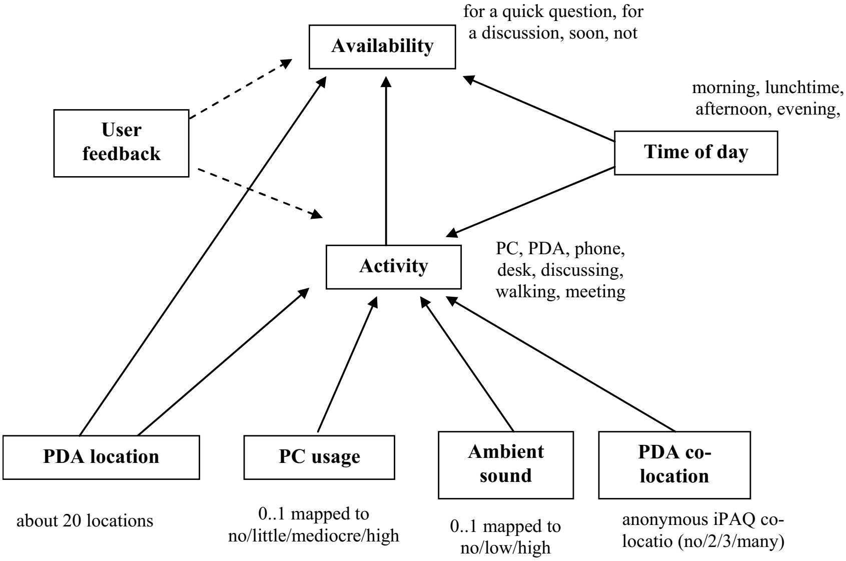

Availability
for a quick question, for a discussion, soon, not
User feedback
morning, lunchtime, afternoon, evening,
Time of day
Activity
PC, PDA, phone, desk, discussing, walking, meeting
PDA location
PC usage
Ambient sound
PDA co-location
about 20 locations
0..1 mapped to no/little/mediocre/high
0..1 mapped to no/low/high
anonymous iPAQ co-locatio (no/2/3/many)

Figure 4. Learning dependencies.

based on the most reasonable situation estimation (see figure 3). The application is also adaptive to changes in a user's environment and habits. Whenever the user provides to the system new samples of information about his activity and availability using the context feedback application, the system can automatically adapt the context estimators and update its situation estimation.

In order to investigate the quality of the situation estimation and to test the sensing infrastructure, several one-day experiments have been conducted with different sets of users, including typical user situations such as PC work, discussing, meeting, etc. After having collected characteristic data during one day, we tried to classify new user-labeled situations the following day. Table 1 and Table 2 show the results of the activity and the availability detection in form of confusion matrices. Each matrix element shows the number of test examples for which the actual class is the row and the predicted class is the column. The training and test sets are comprised of 62 situations (day 1) and 27 situations (day 2), respectively. All situations included the activities "PC, "desk" or "discussing".

The results of the detection of the activities "PC" and "discussing" were very good, because they rely directly on sensor information (PC activity and ambient sound). As the PC activity sensor smoothes its values, it does not immediately return to zero when the user stops working on PC and begins working on desk. That is why there is a quite high detection rate of the activity "PC", even though the user labeled it "desk". The results of the detection of the availabilities "for a discussion" and "not at all" are excellent due to the fact that the users linked these availabilities especially to the time of day during the experiment. Many of them did not want to be contacted in the morning most of the time, but were available for a discussion in the afternoon to a large degree. Furthermore, in the experiments it turns out that a user's location is a strong indicator for his activity. This seems reasonable since in his own room a user typically would be doing PC and desk work, whereas in his colleagues' rooms and meeting rooms he would usually be discussing or meeting, respectively.

	PC	Desk	Discussing
PC	0.74	0.05	0.16
Desk	0.33	0.67	0.00
Discussing	0.00	0.00	1.00

Table 1. Confusion matrix for activity.

	For a quick question	For a discussion	Soon	Not at all
For a quick question	0.50	0.50	0.00	0.00
For a discussion	0.00	1.00	0.00	0.00
Soon	0.00	0.09	0.91	0.00
Not at all	0.00	0.00	0.00	1.00

Table 2. Confusion matrix for availability.

The automatic generation of probabilistic models of human behavior has also been done in other projects. Bayesian learning has extensively been used in the Microsoft Coordinate project for instance to predict peoples' presence at their desks or their interruptibility while being in a meeting [4]. In addition to Bayesian learning, other probabilistic methods have been used to learn and detect human activity, such as an approach based on hierarchical hidden Markov models to learn the hierarchical structure of sequences of human actions [7], although with a different objective, i.e., the extension of the functional capability of the elderly.

4. Summary

The combination of learning group formation based on information from learner profiles and information on the learner context has a potential of improving the quality of the grouping. It allows for the ad-hoc creation of learning groups, which is especially useful for peer help for immediate problems, by reducing the risk of disruptions. It also leverages the forming of face-to-face learning groups based on the presence information. The context sensing has been tested with a set of experiments, and a distributed application has been developed that helps teachers to form learning groups.

Potentially, other context information can be used to improve the group formation than the one that has been considered here, such as agenda information from personal calendars, or the availability of preferred communication channels. The building of learning groups could also be enriched by information available on the experience from past collaborations, which could be provided by peers but also from a teacher if available. Furthermore, in addition to the topic of the collaboration, the group formation could include information on the type of support needed, among others.

Acknowledgement

The work presented in this paper is partially supported by European Community under the Information Society Technologies thematic area of the 6th Framework Programme by the RTD project iClass (IST-507922).

References

[1] Andreoli, J.-M., Castellani, S., Fernstrom, C., Grasso, A., Meunier, J.-L., Muehlenbrock, M., Ragnet, F., Roulland, F., & Snowdon, D. (2003). Augmenting offices with ubiquitous sensing. In *Proc. of Smart Objects Conference SOC-2003*, Grenoble, France, May.

[2] Bahl P. & Padmanabhan, V. N. (2000). Radar: An in-building RF-based user location and tracking system, In *Proc. of the IEEE Infocom-2000*, Tel-Aviv, Israel, vol. 2, Mar. 2000, pp. 775-784.

[3] Hoppe, H. U. 1995, The use of multiple student modeling to parameterize group learning, In J. Greer (Ed), *Proceedings of AI-ED 95*, Washington D.C., USA.

[4] Horvitz, E. Koch, P. Kadie, C.M., Jacobs, A. (2002) Coordinate: Probabilistic Forecasting of Presence and Availability,. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, Edmonton, Alberta, Aug.

[5] Inaba, A., Supnithi, T., Ikeda, M., Mizoguchi, R., & Toyoda, J. (2000). How Can We Form Effective Collaborative Learning Groups?, *Proceeding of ITS 2000*, 282-291, Montreal, Canada.

[6] Jermann, P., Soller, A., & Mühlenbrock, M. (2001). From mirroring to guiding: A review of the state of art technology for supporting collaborative learning. In P. Dillenbourg, A. Eurelings, & Kai Hakkarainen, editors, *Proceedings of the European Conference on Computer-Supported Collaborative Learning, EuroCSCL-2001*, p. 324-331. Maastricht, The Netherlands, March.

[7] Lühr, S., Bui, H.H., Venkatesh, S., West, G.A.W. (2003) Recognition of Human Activity through Hierarchical Stochastic Learning. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications PerCom-03*, 416-421. Fort Worth, Texas, USA

[8] McCalla, G. I., Greer, J. E., Kumar, V. S., Meagher, P., Collins, J. A., Tkatch, R. Parkinson, B., 1997, A peer help system for workplace training, In Boulay & Mizoguchi, editors, *Proceedings of the Conference on Artificial Intelligence in Education AIED 97*, pages 183-190, Kobe, Japan.

[9] Mühlenbrock, M., Brdiczka, O., Snowdon, D., and Meunier, J.-L. (2004). Learning to detect user activity and availability from a variety of sensor data. In *Proceedings of the Second IEEE Conference on Pervasive Computing and Communications*, Orlando, FL, March.

[10] Mühlenbrock, M., Tewissen, F. & Hoppe, H. U. (1998). A framework system for intelligent support in open distributed learning environments. *International Journal of Artificial Intelligence in Education*, 9, 256-274.

[11] Salber, D., Dey, A., and Abowd, G. (1999). The Context Toolkit: Aiding the Development of Context-Enabled Applications. In *Proceedings of the 1999 Conference on Human Factors in Computing Systems CHI 99*, pages 434-441, Pittsburgh, PA, May.

[12] Weiser, M. and Brown, J. S., *Designing Calm Technology*, 1995.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Evaluating Inquiry Learning Through Recognition-Based Tasks

Tom Murray, Kenneth Rath[†], Beverly Woolf, David Marshall,
Merle Bruno[††], Toby Dragon, Kevin Kohler, Matthew Mattingly

University of Massachusetts, Amherst, MA
[†] Peterfreund Associates, Amherst, MA
[††] Hampshire College, Amherst, MA
contact: tmurray@cs.umass.edu

Abstract. The Rashi inquiry learning environment for human biology was evaluated using a new instrument for assessing gains in scientific inquiry skills. The instrument was designed to be sensitive to the small pre-post skill gains that are hypothesized for short learning interventions. It is also designed to be scored with less effort than the verbal protocol analysis methods most often used to asses higher order skills. To achieve these ends the instrument is "item-based", "recognition-based" and "difference-based." We describe our assessment design method and results of its first use.

1. Introduction

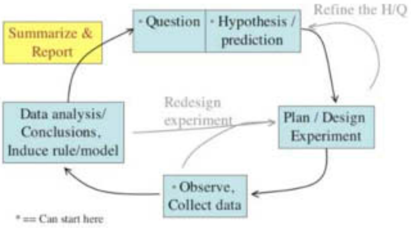
Rashi is a domain independent architecture for inquiry learning environments. It contains tools that allow learners to gather data, pose multiple hypotheses, and create arguments that support hypotheses by linking to supporting or refuting data. We are using Rashi to build inquiry learning environments in human biology, geology, and forest ecology, all for undergraduate level science. Though inquiry skills, like all higher order thinking skills, are difficult to assess [1], it is important that we develop methods for assessing these skills because they are essential in many types of work and problem solving, and they are given high priority in many educational standards and frameworks.

A common problem in research into advanced learning environments is that the software is not able to be tested in authentic contexts over extended periods of use. Such systems usually have significant pedagogical "depth" but little content scope, and when they are employed in classrooms their content applies to a very small portion of the curriculum. Also, it may be difficult to find instructors willing to "give up" significant course time to an alternative approach. The fact that our interventions may be limited to weeks or even hours is at odds with the slow rate of improvement expected for higher order cognitive skills. In order to evaluate these interventions instruments need to be sensitive to small learning gains.

In this paper we describe our first attempts with a new methodology for developing assessments for inquiry learning environments. Our goals are to design inquiry assessment instruments that are: 1) sensitive to small changes in skill level, and b) less labor intensive than most currently used methods. The method uses recognition-based (as opposed to recall), item-based (as opposed to free-form), and difference-based tasks (as described later). We describe our first use of this method, its results, and planned improvements on the method. Data analysis of the results revealed no statistically significant conclusions, and this we attribute to a non-optimal subject context (there was insignificant motivation for the volunteers to take the task seriously) which will be avoided in future trials. Thus the contribution of this paper is more in the description and discussion of the methodology than about evaluation results.

2. An Inquiry Learning Model

Our model of inquiry learning is based in part on knowledge from the experts we work with, and in part on the inquiry-based learning literature ([2],[3],[4], among others). Figure 1 shows our model of the "scientific inquiry process," which combines elements from other models.



The following inquiry skills have been identified as most important by our subject-matter experts:

Table 1

1. **Understand** the task and what constitutes completion of the task
2. **Differentiate** observation (and data) from inferences
3. **Justify** hypotheses with arguments
4. **Explain** inferences and hypotheses
5. **Explore** observation, measurement, and information resources
6. **Cite** source documents
7. **Systematically** gather, interpret, and organize information.
8. **Communicate** a clear summary of your findings in written form.

We have used this skill list to inform the design of the Rashi tools, and to inform the design of our evaluations.

3. Assessment design issues

We will describe three methodological decisions which resulted in our assessment task being "item-based," "recognition-based," and "difference-based."

Item-based tasks. The most common methods used in researching inquiry-based, discovery-based, or open-ended learning environments are qualitative and ethnographic methods. Such methods include analysis of verbal data from peer work or structured interviews and analysis of written work from assignment portfolios or journals. They are appropriate for interpretive research aiming for a "thick" characterization of the student/user experience for a small number of cases, and are very labor-intensive. The literature includes many examples of inquiry-based educational technology projects that have used such evaluation methods ([2],[5],[6],[7],[8],[9]). In exploratory research where the key questions and constructs are still being worked out, ethnographic methods are used because they allow the nature of the data analysis to evolve during the analysis. But when a theoretical framework already exists, more specific types of tasks can be designed. These "closed" tasks, such as sorting, ranking, and comparison tasks (see [10],[11]), can be more reliable and generalizable, but tend to have less authenticity, which may effect the ecological validity of results.

As a compromise between closed tasks and more open-ended tasks, case data (verbal or written) should have clear segmental boundaries such as answers to questions or problem solution steps, allowing many data points per subject (see "intra-sample statistical analysis" in [12]). We will call this approach item-based to refer to segmenting the task into discrete task items. Scoring rubrics can then address more constrained tasks ([13],[14]).

Recognition-based tasks. As mentioned, significant gains in higher-order thinking skills usually require significant learning time. Yet for these trials we were limited to 2 or 3 sessions of 2 to 3 hours each. Given these constraints, we hypothesized that a recognition task would be more likely to show skill improvement than a recall task. Recognition learning usually occurs more easily than recall learning. For example, imagine that someone reads a list to you and then reads from another list and asks whether or not each of the items was on the original list. This task is easier than trying to recall all of the items from the original list without being given any cues. Our recognition task for inquiry learning skills involves rating the quality of a hypothetical problem solution, rather than generating a problem solution from scratch.

Roth & Roychoudhury [3] discuss the importance of a socio-cultural perspective on teaching scientific inquiry, noting that "new and more powerful skills and concepts can be observed in social interaction long before they are exhibited by individuals" pg (133). The collaborative context can be particularly useful to exposing learning gains. We propose that this will still be the case (thought to a lesser degree) in "mock" sociological contexts, as in when a student is asked to critique or rate the work of a hypothetical peer.

Difference-based tasks. The third methodological decision was that the post-test task involved asking subjects to *improve* upon their pre-test answers as opposed to solving an entirely new problem. We believe that this "difference-based task" will further sensitize the instrument to small gains in inquiry skills. It also removes some of the variability introduced when using different pre-and post test tasks that have not undergone rigorous psychometric verification of equivalence.

Before further describing the instrument we will briefly describe the software evaluated.

4. Description of Biology Domain and Rashi Tools

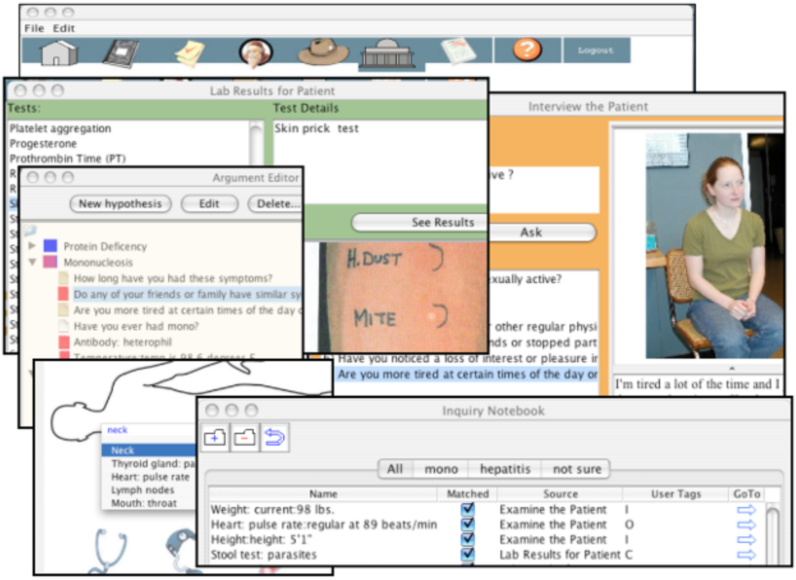
Rashi domains and inquiry tasks. In the Rashi Human Biology (HB) Tutor, learners are presented with medical cases and attempt to diagnose the patient's condition. They can interview the patient, perform various physical exams, and order lab tests for blood, urine, etc. Rashi includes an authoring tool [15] that allows authors to create new cases. The Rashi HB tutor is based on a case-based classroom teaching method used by one of the co-authors at Hampshire College [16]. Eight cases have been authored from Rashi HB, based on medical conditions including mold allergies, hyperthyroidism, and lactose intolerance.

Next we give a very brief overview of the tools available to learners in the Rashi system (and see [17],[18],[19],). Rashi provides a set of tools that map onto the inquiry skills mentioned in Table 1 .

- **Case Orientation Screen**: Provides information about the case and general problem solving instructions. (Supports skill #1 in Table 1.)
- **Data gathering tools**: Each domain has its own set of data gathering tools. For the human biology domain they include a patient interview, physical exam, and lab tests. (Supports skill #5)
- **Inquiry Notebook**: Gathered data is saved to the inquiry notebook, which allows the setting of the data source, confidence level, and data type (hypothesis, measurement, observation, etc.) for each item. Data can be organized into folders (like having different pages in a research notebook), and keyword tags can be entered for sorting the items. (Supports skills #2, 6, 7)
- **Argument Editor**: Users create hypotheses and create arguments for and against them through links to notebook data items. Hypotheses are rated (e.g. top, possible, ruled out), and the argument relationship types are specified (e.g. supports, refutes, etc.).

Users can enter explanations for their hypotheses and for each argument link. (Supports skills #3, 4)

Rashi includes a Planning Scratch Pad (for skill #7), a Sources Editor (for skill #6), a Concept Library (for skill #5), and a Reporting Tool (for skill #8). The figure below shows some of the tools from the Biology domain (lab test results in upper left; patient interview in upper right, physical exam in lower left, argument editor in middle left, and notebook in lower right, with the main screen showing icons to access the tools shown at the very top). Rashi also has an intelligent coach, but this was turned off for these studies because the advice it gives was not yet robust enough. Also, we wanted this study to serve as a baseline for evaluating the system with the coaching turned on.



Name	Matched	Source	User Tags	GoTo
Weight: current:98 lbs.	☑	Examine the Patient	I	⇨
Heart: pulse rate:regular at 89 beats/min	☑	Examine the Patient	O	⇨
Height:height: 5'1"	☑	Examine the Patient	I	⇨
Stool test: parasites	☑	Lab Results for Patient	C	⇨

5. Methodology

Evaluation context and goals.

In the Fall of 2004, we evaluated inquiry skill gains resulting from Rashi HB use in two college classrooms. The first trial was in Bruno's small introductory Biology class, and served as a pilot test of our inquiry skills instrument. The second marked the first time Rashi had been used in the context of a large lecture class.

Having developed and tested Rashi in the context of a small-sized classroom with a teacher skilled in case-based inquiry pedagogy (which is not the context in which we expect it to show the largest benefit over the usual classroom experience), we wanted to test the system in the context of a larger classroom where the instructor did not have a high level of inquiry teaching skill.

Unfortunately, we were unable to find a large-sized college class in Fall of 2004 where the Rashi activities could be integrated, but we found an large introductory biology class for which the Rashi activities could be assigned as *extra credit*. The amount of extra credit time available was limited to 6-8 hours, including, instructions, and survey/test-taking.

Evaluation instrument

Developing evaluation tasks and instruments for inquiry learning environments is still very much a "black art," so below we describe in some detail how we developed ours. The Rashi tools are designed with a specific inquiry task model in mind, and we designed our evaluation task according to this model. As mentioned, the evaluation task involved presenting the subject with a hypothetical (and "imperfect") case solution created by an imaginary "student investigator", and asking the subject to evaluate its quality.

A. **Task design**. We wanted the evaluation task structure to parallel the task structure of using the Rashi tools to solve a case, so be broke up the Hypothetical Case Solution into three parts roughly corresponding to the main Rashi Tools. Solution Part A ("Beginning the Case") consisted of lists or initial hypotheses and what information is needed to confirm or reject them. Part B ("Data Collection") consisted of a list of data collected, with reasons. Part C ("Diagnosis Justification") consisted of a final set of accepted and rejected hypotheses, with justifications pointing to the data collected.

For all three parts of the pre-test, the instructions said: "List at least two strengths and two weaknesses of the investigator's notes." For the post-test, subjects were given exactly the same exercise and a copy of their previous answers. The only difference was the instruction to look at their pre-test answers and list at least one additional strength and weakness of the investigator's notes.

B. **Ideal Solution Characteristics**. We developed a model solution rubric describing the characteristics of a "correct" set of investigator notes for the task. We developed this list from the list of inquiry skills and through piloting the instrument and looking at the types of correct and incorrect statements that students made.

C. **Case Creation**. We created a case that focused on a different medical topic than that used in the Rashi software. The Case Description given to subjects included:
"Jean Rockford, a 26-year-old woman, comes to see you with a 6-month history of increasing nervousness, irritability, and heat intolerance...."

D. **Ideal Solution Instance**. We constructed an ideal diagnosis solution, including approximately 15 items for each of the three parts, which included all of the characteristics of an ideal solution.

E. **Imperfect Solution**. We modified this ideal solution to create the final Hypothetical Case Solution with errors of omission and commission. This was a delicate "operation" because we felt the final investigator notes should have a range of easy-to-notice to difficult-to-notice errors geared to differentiate skill levels. In addition, the entire set of Investigator's notes had to look reasonable, being mostly correct but with a tractable number of identifiable problems.

F. **Scoring Rubric Development**. Finally we developed a scoring rubric geared for the specific case. The "imperfect solution" had a total of 16 faults, for a total of 16 possible points in the "list the weaknesses" questions (the "strengths" questions were not scored).

Experimental method

Experimental and control groups. In addition to the Rashi-using experimental group, we had three additional comparison groups, named according to the task given: Non-interactive case investigation, Inquiry article reading task, and Biology article reading task. These groups were created to allow credit assignment for any gains observed in the Rashi-using group (i.e. to attribute such gains to the interactive software, or the case-based instructional method, or to an exposure to inquiry concepts).

The **Rashi** Group used the Rashi system to investigate a medical case. The **Non-interactive** Group was given the same medical case to diagnose, but instead of using the Rashi system for their investigation they used a web site with static information about the case and were given worksheets with tables for keeping track of "things I need to know," "data gathered" and "diagnostic hypotheses." Both the Rashi group and the Non-interactive-inquiry group were asked to write up a 1-3 page summary report of their investigation and conclusions, and email this to us. The **Inquiry-reading** Group was given an article about using inquiry learning methods in science, and the **Biology-reading** Group was given a research article on diet's relationship to cardiac illness. Both reading groups were asked to write 1-3 page summaries of the articles and email them to us.

We hypothesized that inquiry learning improvements in the four groups would be ordered as: Rashi > Non-interactive task > Inquiry-reading > Biology-reading. Our reasons were as follows. The more realistic and interactive features of Rashi, plus the tools it gives students to organize and visualize information, should have helped students focus on their inquiry process and thus improve skills, as compared with the non-interactive task. Constructivist learning theory predicts that the two inquiry tasks would fare better than the two reading tasks. Also, we expected that reading an article about inquiry learning might have a slight effect on students, while reading an article on an unrelated topic should not.

Additional measures

Software use records. Our software currently stores all student work on a central server, but does not record each student action as they are using the Rashi tools. For this study we compiled a number of feature-use statistics based on the final state of the subject's work

Attitude Survey. The students in the Rashi Group filled out a survey appended to the on-line post-test. The survey included a 11x3 response matrix where the 11 rows listed activities or skills that the software supports (e.g. understanding the entire inquiry process, gathering data and information, citing the sources of information) and the columns asked: A. "How *successful* were you at the following activities"; B. "How *easy* was it for you to do these activities"; and C. "How *important was Rashi* in your ability to do these activities." For each of the 33 cells in the response matrix, students selected from three Likert-scale values. In addition, subjects were asked how much time they spent on the Rashi task.

Experimental Context

Volunteers from an undergraduate biology class of about 500 students were offered extra credit for participating in the study. Of the 140 students who signed up and began the processes, only 74 finished all required tasks. The number of students who completed all tasks, along with average self-reported time, is shown below.

Group	N	Time to Complete
Rashi	17	2.4 hours
Non-interactive-inquiry	18	2.5 hours
Inquiry-reading	17	2.4 hours
Biology-reading	22	2.0 hours

6. Results and Analysis

The following table gives the average test scores, their difference, and the standard deviation of that difference, t-test, and significance for each of the four groups.

Group	N	Average Pre	Average Post	Average Diff	SD Diff	t-test	p
Rashi	17	0.71	1.00	0.29	0.47	2.58	0.020
Non-interactive	18	1.00	1.61	0.61	0.78	3.34	0.004
Inquiry-reading	17	0.94	1.29	0.35	0.61	2.40	0.029
Biology-reading	22	0.36	0.59	0.23	0.43	2.49	0.021
Total	**74**	**0.73**	**1.09**	**0.36**	**0.59**	**5.35**	**0.000**

The results indicate an extreme floor effect (with average pre and post tests scoring about 1 out of a possible 16 points). There was no significant differences between groups on any measure. An ANOVA analysis of the results found that there were no statistically significant differences in the amount of improvement on inquiry skills across the four groups (F(3, 70) = 1.58, p = 0.20). The effort given by students in all four groups is similar (2 to 2.5 hours), though we expected students in the two inquiry tasks to spend significantly more time than they did on the task. (Note: Because the of the difference-based nature of the post-task, we would expect all post-tests to have higher scores then pre-tests, thus the low p values.) Combining the first two groups into an "inquiry-based" set and the last two into a "reading-based" set and comparing inquiry-based with reading-based also shows no significant differences .

Attitude survey results. As in past formative evaluations of Rashi, the survey did not indicate any significant problems with the software. We interpret these results as supporting the usability of the software and its perceived usefulness, especially given the short amount of time students were introduced to it and used it, and the fact that the study task did not relate to their current classroom activities.

Software use metrics. Since there were no significant differences between the pre- and post tests, we will call the subject's pre-test score their "inquiry skill level." There were significant correlations between inquiry skill level and some of the Rashi use metrics. In particular, there were significant positive correlations between inquiry skill level and the number of hypotheses posed, the number of arguments, the number of items in the notebook, the number of explanations entered by students, the use of notebook organizing tools, and the overall use of Rashi tools. As this is what one would expect, this adds some credence to the ecological validity of the pre-post instrument.

7. Discussion of Results

Floor-effect. As mentioned, our evaluation suffered from a significant floor effect, which makes it difficult to compare results of the four experimental groups. Some of this can be attributed to the design of the instrument, but we believe that mostly the floor effect is a result of characteristics of the subject population. We believe that the subjects were not motivated to take the study very seriously and put the necessary mental effort into the evaluation and intervention tasks. We believe that this was because: 1) the tasks were not integrated into the classroom experience and had nothing to do with content covered in the class; 2) volunteers signed up only to receive extra credit, and did not take the evaluation tasks very seriously because they were only required to complete the steps of the study to receive extra credit.

Improvements. We plan to carry out evaluations of Rashi in about 5 classrooms in 2005. Improvements based on lessons learned from the current study will include: 1) clearer pre-post test instructions to focus subjects on inquiry-specific skills; 2) rewording the "2 or more" strengths and weaknesses questions to encourage more answer items; 3) performing the study in classrooms that have the intervention activities more integrated into classroom activities.

8. Conclusions

This study did not yield very informative results due to floor effects, which in the future should be remedied by one or a combination of the improvements mentioned above. However, we believe that our suggestions for the development of assessment instruments are innovative in the context of assisting inquiry learning environments, and worth pursuing further.

To summarize, our goals were 1) to develop an instrument sensitive to changes in inquiry skills after relatively brief interventions, and 2) to develop an instrument that could be scored with relatively little effort. We believe that we succeeded on the second point, since the scoring of all 74 pre and 74 post tests was done by one person within a single day.

Our methods for developing more sensitive instruments for inquiry skill included creating an assessment task that was "recognition-based," "item-based," and "difference-based," as described above. Due to the difficulties with the present study, we do not know yet whether these methods are in fact useful. Our further studies in 2005 will answer this question.

A further methodological innovation was that we used system tracking data along with skill assessment and survey data, which is rarely done in studies of inquiry learning systems. This allows us to construct more elaborate explanations for any significant differences we find within or between experimental groups. Our method of constructing a comparison task starting with ideal solution characteristics based on the inquiry model, then creating an ideal solution, and then perturbing the ideal solution to create the final imperfect Hypothetical Case Solution also seems unique to inquiry learning environment evaluations.

References

[1] Champagne, A.B., Kouba, V.L., & Hurley, M. (2000). Assessing Inquiry. In J. Minstrell & E. H. van Zee (Eds.) *Inquiry into Inquiry Learning and Teaching in Science*. American Association for the Advancement of Science, Washington, DC.

[2] White, B., Shimoda, T., Frederiksen, J. (1999). Enabling students to construct theories of collaborative inquiry and reflective learning: computer support for metacognitive development. *International J. of Artificial Intelligence in Education* Vol. 10, 151-1182.

[3] Roth, W. & Roychoudhury, A. (1993). The Development of Science Process Skills in Authentic Contexts. *J. of Research in Science Teaching*, Vol. 30, No 2. pp. 127-152.

[4] Edelson, D.C., D.N. Gordin, and P.D. Pea (1999). "Addressing the Challenges of Inquiry Based Learning Through Technology and Curriculum Design." The *Journal of Learning Sciences*. 8(3&4): 391-450. 1999..

[5] Azevedo, R., Verona, E., Cromley, J.G. (2001). Fostering learners collaborative problem solving with RiverWeb. J.D. Moore et. al. (Eds.) *Proceedings of Artificial Intelligence in Education,* pp. 166-172.

[6] Krajcik,J., Blumenfeld, P.C., Marx, R.W., Bass, K.M., Fredricks, J. (1998). “Inquiry in Project-Based Science Classrooms: Initial Attempts by Middle School Students” *J. of the Learning Sciences*, 7(3-4), pp 313-350, 1998.

[7] van Joolingen, W., & de Jong, T. (1996). Design and Implementation of Simulation Based Discovery Environments: The SMILSE Solution. *Jl. of Artificial Intelligence in Education* 7(3/4) p 253-276.

[8] Zachos, P., Hick, T., Doane, W., & Sargent, C. (2000). Setting Theoretical and Empirical Foundations for Assessing Scientific Inquiry and Discovery in Educational Programs. *J. of Research in Science Teaching* 37(9), 938-962.

[9] Murray, T., Winship, L. , Stillings, N. (2003B). Measuring Inquiry Cycles in Simulation-Based Leaning Environments. Proceedings of Cognitive Science, July, 2003, Boston, MA.

[10] Mestre J. P. (2000). Progress in Research: The interplay among theory, research questions, and measurement techniques. IN Lesh (ed)..

[11] Toth, E.E., Klahr, D, Chen, Z. (2000). Bridging research and practice: A cognitively based classroom intervention for teaching experimental skills to elementary school children. *Cognition and Instruction*, 18(4), 423-495.

[12] Shaffer, D.W. & Serlin R.C. (2005). What good are statistics that don't generalize? *Educational Researcher* 33(9), 14-25.

[13] Lunsford, E. & Melear, C. T. (2004). Using Scoring Rubrics to Evaluate Inquiry. J. of College Science Teaching 34(1), 34-38.

[14] Stillings, N. A., Ramirez, M. A., & Wenk, L. (1999). Assessing critical thinking in a student-active science curriculum. Paper presented at the meeting of the National Association of Research on Science Teaching, Boston, MA.

[15] Murray, T., Woolf, B. & Marshall, D. (2004). Lessons Learned from Authoring for Inquiry Learning: A tale of authoring tool evolution. J.C. Lester et al. (Eds.). ITS 2004 Proceedings, pp 197-206.

[16] Bruno, M.S. & Jarvis, C.D. (2001). It's Fun, But is it Science? Goals and Strategies in a Problem-Based Learning Course. J. of Mathematics and Science: Collaborative Explorations.

[17] Murray, T., Woolf, B., Marshall, D. (2003). Toward a Generic Architecture and Authoring Tools Supporting Inquiry Learning. Proceedings of AI-ED'2003, 11th World Conference on Artificial Intelligence in Education, 20-24 July, 2003. Sydney, pp. 488-490.

[18] Woolf, B.P., Marshall, D., Mattingly, M., Lewis, J. Wright, S. , Jellison. M., Murray, T. (2003). Tracking Student Propositions in an Inquiry System. Proceedings of AI-ED'2003, 11th World Conference on Artificial Intelligence in Education, 20-24 July, 2003. Sydney, pp. 21-28.

[19] Woolf, B. P. et. al [in submission]. Critical Thinking Environments for Science Education.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Personalising information assets in collaborative learning environments

Ernest Ong, Ai-Hwa Tay, Chin-Kok Ong, Siong-Kong Chan
Institute of Systems Science, National University of Singapore
25 Heng Mui Keng Terrace, Singapore 119615
Email: ernie@iss.nus.edu.sg, AiHwa.Tay@seagate.com, ChinKok@techsemicon.com.sg,
Siong-Kong.Chan@reedhycalog.com

Abstract. The volume of information in collaborative learning environments can be daunting. Towards the objective of providing the learner efficient access to knowledge by applying knowledge management practices, we posit the use of topic maps, an ISO standard for structuring and indexing information, to support knowledge organisation (KO). We suggest how KO and Bayesian techniques can support collaborative learning to enable more efficient organisation of and access to knowledge artifacts arising from collaborative interactions. To test these ideas, we implemented a prototype called Adaptive Recommendation Module (ARM) for use within our production information portal Knowledge@Work.
Keywords: Knowledge organization, e-learning, computer-supported collaborative learning (CSCL), topic maps, Bayesian inference, web mining, personalisation

1. Introduction

The constructivist learning approach is often criticised for its lack of well-defined context within which progressive inquiry can take place [5,8]. In response to this, many computer-supported collaborative learning (CSCL) environments have used note-taking as one of the primary means by which peers produce knowledge building artifacts and engage in interactions in shared working spaces.

In our recent work [1], we applied knowledge organization (KO) techniques and topic map technologies [4,6] to organise and to manage efficient access to dynamic communal knowledge [7]. To facilitate efficient access to information in such contexts, one must first address the issue of semantic interoperability - the comparability and the compatibility of knowledge structures - when organising and integrating metadata. These challenges are not unlike those faced by CSCL environments where peers contribute to the collective learning experience and cope with the task of managing, presenting and reconciling the multiple perspectives.

Designing efficient knowledge structures is expensive. This is especially so when the body of information assets is expansive and continually evolves. Consequently, such knowledge structures are subsequently reorganised incrementally rather than substantially. Ong, Looi and Wong [3] proposed *organic knowldege maps* as a means to efficiently manage dynamically evolving communal knowledge. This was implemented within a web information portal Knowledge@Work (http://www2.iss.nus.edu.sg/portal) that is used to facilitate collaborative online interactions as part of the blended learning experience at our institute. Knowledge artifacts such as discussions, personal notes and knowledge maps can be re-purposed and re-organised to create new, sharable knowledge maps. These knowledge maps then form the basis for spawning new conversations and further knowledge maps.

One important limitation of the above work is the lack of mechanisms to manage the signal-to-noise ratio when presented with a vast volume of information assets and to present an organised view of such assets. To this end, there are two popular complementary approaches. The first involves augmenting the portal navigation with presence information: who is viewing what and where these portal assets are in real-time. This model is particularly suited to usage scenarios where highly proactive users are spending substantial online time simultaneously. The second involves periodic analysis of usage patterns and recommending portal assets which may be of relevance and interest to the user. This model is more suited to usage scenarios with insubstantial overlapping online time.

Based on observed usage patterns on Knowledge@Work, we decided to explore the latter approach which is the subject of this paper. The prototype Adaptive Recommendation Module (ARM) uses a combination of techniques including user profile matching, probabilistic reasoning, Bayesian inference and topic maps to determine the relevance of information assets.

This paper is structured as follows. Section 2 describes the personalisation model. Section 3 describes the functional modules and scoring algorithm. Section 4 provides an overview of initial test results and finally we conclude in Section 5 with a discussion of applicable usage scenarios.

2. Personalisation model

Mindful of the extensive effort required for comprehensive design exercises, the primary challenge was to minimise the involvement of subject matter experts (SMEs) during the metadata tagging phase; this represents the *static view* from the experts' perspective. Furthermore, the resulting metadata has to be amenable to support analysis of usage logs; this allows *dynamic changes* to be incrementally introduced based on the analysis of observed behaviour. The critical link between the static (based on beliefs) and the dynamic (based on actual usage) is the personal *user profile*. The remainder of this section describes this model in greater detail.

2.1 Structural information (static)

To simplify the task of the SME, only three layers of metadata are required (Figure 1). The first describes the overall structure eg. Java; Java → J2EE; Java → J2EE → EJB. These non-terminal nodes, representing categories, are containers for information assets. The second layer describes the information assets or terminal nodes in the form of content articles. Both terminal and non-terminal nodes are also known as *topics*. This layer also describes the relationship or *associations* between information assets and non-terminal nodes and, optionally, other information assets. Finally, the third and final layer, identifies the asset instance, or *occurrence* in topic map parlance, for each information asset.

Topic maps afford great flexibility in how information assets are managed and structured. However, the extensive and often technical vocabulary of topic maps can be daunting to the average SME thus posing usability and productivity problems. To reduce the complexity of the ontology imposed on SMEs, a customised vocabulary was introduced along with some simplifications. Additional content guidelines further help SMEs design the category structure of Layer-1. For example, the nesting relationship denotes "is-part-of" specialisation. Consequently, information assets in Java → J2EE are more general than those in Java → J2EE → EJB. SMEs then assign appropriate belief values (subjective probabilities) expressing the relevance of each category to different user proficiencies. Four belief values are assigned, one for each of the four user proficiency levels – novice, intermediate, advance and expert. These subjective values denote the degree to which a user with the given proficiency level might be interested in the category.

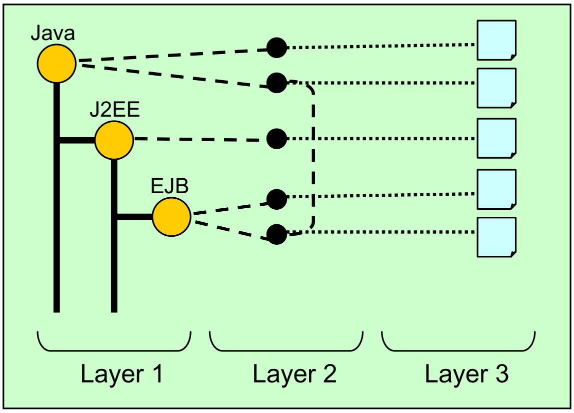


Figure 1: Topic maps and structural information

Definition 1 (association)

Let **u** and **v** are two topics (category or information asset) in a topic map, and **t** a valid association type within the topic map. The topic **u** is said to have a dominant association of type **t** with respect to **v** and is written $\mathbf{v} \overset{\mathbf{t}}{\leftarrow} \mathbf{u}$. The association type **t** may be omitted and the simplified expression is written $\mathbf{v} \leftarrow \mathbf{u}$. In Figure 1, the associations between Java, J2EE and EJB may be written EJB ← J2EE ← Java.

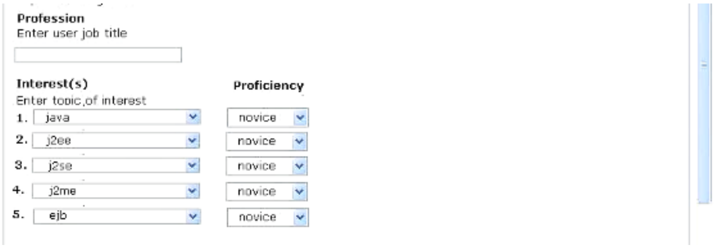
Likewise, each information asset in Layer-2 is tagged albeit with numerical values representing the nearest proficiency level of the "ideal" target user for the information asset (Figure 2).

Proficiency level	Value range
Novice	1.0 to 1.9
Intermediate	2.0 to 2.9
Advance	3.0 to 3.9
Expert	4.0

Figure 2: "Ideal" proficiency level for information assets

2.2 Personal user profile (preference)

Users can indicate the categories, defined in Layer-1, of interest to him (Figure 3). For each category, the user indicates his proficiency level which is interpreted numerically as follows: novice as 1.0, intermediate as 2.0, advance as 3.0 and expert as 4.0. We are aware of concerns regarding explicit data acquisition including privacy and data integrity [2]. The latter could be addressed to some extent via computed proficiency levels based on externally gathered data and peer feedback eg. assessment grades; peer rating of artifacts.



Profession

Enter user job title

Interest(s) Proficiency

Enter topic of interest

1. java novice
2. j2ee novice
3. j2se novice
4. j2me novice
5. ejb novice

Figure 3: User profile preferences

Definition 2 (proficiency preference)

The proficiency preference **pref** for user **u** is a partial function from categories (non-terminal topics) to natural numbers ℕ.

$$\mathbf{pref}_{\mathbf{u}} : \textit{topic} \nrightarrow \mathbb{N}$$

2.3 Collaborative filtering (dynamic)

The structure of the topic map representing categories and information assets is mostly static. SME involvement is required only during periodic updates, for example when adding new categories or information assets. This significantly reduces the cost of running an information portal. However, this also restricts the degree of personalisation which is based solely on static metadata supplied by SMEs. Relying entirely on the knowledge and experience of SMEs is undesirable for several reasons. Firstly, the performance across SMEs may not be consistent, owing to different levels of experience and expertise, and therefore highly subjective. Furthermore, encoding an exhaustive set of cross-relationships between categories and information assets is not tractable due to cost and the subjective nature of knowledge.

Collaborative filtering is used to mitigate the problem of inconsistent, incomplete and inaccurate metadata. Information assets, including evolving knowledge artifacts [3], which would otherwise have been excluded, may be recommended to a user based on the behaviour of a set of users with similar profiles. However, collaborative filtering requires a sizeable body of usage statistics to provide accurate recommendations [2]. In the absence of reliable usage statistics, the subjective belief values supplied by the SME are used (see Section 2.1).

3. Functional modules

The ARM recommendation engine can be invoked in various contexts to retrieve a sequence of ranked information assets relevant to the respective context. For example, ARM can be used to recommend information assets relevant to the user when navigating structured categories in an information portal or when viewing information assets. The engine is also highly suited as a navigational aid when browsing knowledge maps [3], exposing contextually relevant artifacts representing alternative and possibly new perspectives. The remainder of this section describes the scoring strategy used in ARM.

3.1 Structural distance

The structural distance is computed from the current category with an increment of 0.5 for each edge-traversal through the topic map (Figure 4). A larger increment may unfairly penalise moderately distant assets. In situations where there are multiple paths to the same node in the topic map, the shortest distance is used. The final result is incremented by 1.0 to ensure a non-zero minimum value.

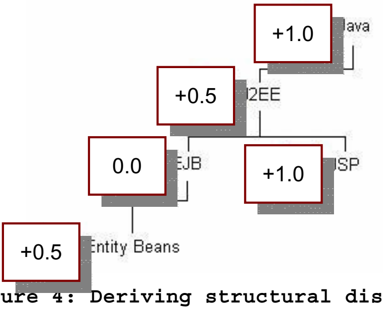


Figure 4: Deriving structural distance

Definition 3 (ancestor)
Let **u** and **v** are two non-terminal topics (categories) in a topic map. We say that **v** is an ancestor of **u** if there is a set of associations **u** ← ... ← **v** in the topic map, written **u** ←* **v**. In Figure 4, the relationship between Java and EJB may be expressed as EJB ←* Java.

Definition 4 (edge-count)
Let **u** and **v** are two non-terminal topics (categories) in a topic map. The edge-count operator || **u** – **v** || is the cardinality of the *minimal set* satisfying one of the following properties:

1. The set comprising the associations satifying **u** ←* **v**. That is, the set of associations establishing the ancestry of **v** with respect to **u**.

2. The set comprising the associations satisfying **u** ←* **w** and **v** ←* **w** for some topic **w** in the topic map. That is, both **u** and **v** share a common ancestor **w** in the topic map.

The *structural distance* **dists** of an information asset **a**, where **a** ← **c** for some category **c**, is defined as follows:

$$\mathbf{dists_a} = (\, \| \mathbf{c} - \textit{current-category} \| \times 0.5) + 1.0$$

For example, the Java category has the structural distance of ((2 × 0.5) + 1.0) = 2.0 with respect to EJB in Figure 4.

3.2 User proficiency distance

Next, information assets belonging to categories of interest declared in the user's personal profile, and their ancestors, are considered. The proficiency distance is computed using the user's declared proficiency level **pref$_u$(c)** for the category **c** as the base (Section 2.2). The final result is, once again, incremented by 1.0 to ensure a non-zero minimum value.

The *proficiency distance* **distp** for the information asset **a**, given the category $\mathbf{c_i}$ satisfying **a** ← $\mathbf{c_i}$, the "ideal" proficiency level $\mathbf{l_a}$ for the asset (Section 2.1) and the set of all categories in the user's profile $\underline{\text{dom}}(\mathbf{pref_u})$ for the user **u**, is defined as follows:

$$\begin{aligned}\mathbf{distp_{u,a}} &= |\, \mathbf{pref_u(c_j)} - \mathbf{l_a} \,| + 1.0, \text{ if } \mathbf{c_j} \in \underline{\text{dom}}(\mathbf{pref_u}) \text{ and } \mathbf{c_i} = \mathbf{c_j} \text{ or} \\ &\qquad \text{for some } \mathbf{c_j} \in \underline{\text{dom}}(\mathbf{pref_u}) \text{ where } \mathbf{c_j} \leftarrow^{*} \mathbf{c_i} \\ &= |\, 1.0 - \mathbf{l_a} \,| + 1.0, \text{ otherwise}\end{aligned}$$

3.3 Collaborative filtering using probabilistic reasoning

Finally, the collective experience of users with similar profiles is considered. The belief values are computed for each combination of category and user proficiency level. Where reliable usage statistics are not available – those with usage levels for the category or proficiency level two standard deviations below the mean – the SME assigned belief values are used in concert with Bayes' theorem. Otherwise, conditional probability is preferred.

That is, assuming reliable usage statistics are available, the conditional probability P(*hypothesis* | *evidence*) of a hypothesis given some observable evidence is computed using available data. In our context, this translates into:

P(*user interested in category C* | *user has proficiency level L associated with C*) ≡ P(C | L)

Note that, in the conditional probability $P(C \mid L) = P(C \cap L) / P(L)$, the conjunctive probability is derived from usage statistics (objective). In particular, the frequency with which a user accesses the category C, for which he has the proficiency level L, can be computed from the usage logs. As the volume of activity increases, so does the accuracy of the recommendations. Furthermore, given

$$P(C \cap L) = P(C \mid L) \times P(L) = P(L \mid C) \times P(C)$$

we have Bayes' theorem

$$P(C \mid L) = (\ P(L \mid C) \times P(C)\ ) / P(L)$$

where $P(L \mid C)$ is the subjective belief value assigned by the SME for each category, described in Section 2.1. Bayes' rule is invoked only if reliable data is not available.

The collaborative filtering factor $\mathbf{col}_{\mathbf{u,a}}$ for the user **u** and the category **c** to which the information asset **a** belongs is defined as follows:

$$\mathbf{col}_{\mathbf{u,a}} = 1 - P(\ C{=}\mathbf{c} \mid L{=}\mathbf{pref}_{\mathbf{u}}(\mathbf{c})\ )$$

This factor establishes a link between the actions of the collective, with a profile similar to that of a user, to those which might be of relevant to the individual, dynamically changing as new information is available. On a practical note, the adaptive nature of collaborative filtering obviates the need to maintain pristinely consistent, accurate and complete metadata which requires frequent maintainance. This significantly reduces the cost of running an information portal.

3.4 Ranking information assets

Let **a** be an information asset belonging to the category **c**. Then, given the structural distance $\mathbf{dists}_{\mathbf{a}}$ with respect to the current category, the proficiency distance $\mathbf{distp}_{\mathbf{u,a}}$ and the collaborative filtering factor $\mathbf{col}_{\mathbf{u,a}}$, the score $\mathbf{score}_{\mathbf{u,a}}$ assigned to the information asset is defined as follows:

$$\mathbf{score}_{\mathbf{u,a}} = \mathbf{dists}_{\mathbf{a}} \times \mathbf{distp}_{\mathbf{u,a}} \times \mathbf{col}_{\mathbf{u,a}}$$

When the score has been computed for all information assets, the assets are sorted in ascending order of their scores where lower scores are ascribed higher rankings.

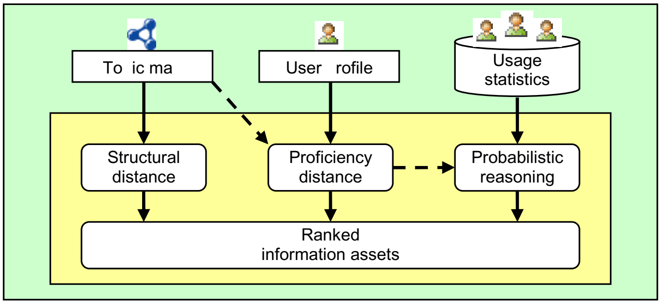
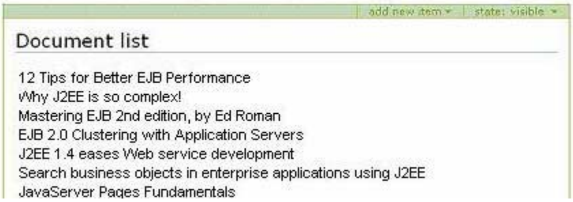


Figure 5: Components of ARM

4. Initial results

For our initial tests, we considered two relatively different user profiles: EJB Expert and JSP Novice. The Mean Squared Error (MSE) index was used to measure the performance of the computed score against the users' target rankings for the top ten information assets ranked by ARM. A user-assigned ranking of twelve denotes strong disagreement, indicating that the information asset should not be included in the list of top ten assets.



Document list

12 Tips for Better EJB Performance
Why J2EE is so complex!
Mastering EJB 2nd edition, by Ed Roman
EJB 2.0 Clustering with Application Servers
J2EE 1.4 eases Web service development
Search business objects in enterprise applications using J2EE
JavaServer Pages Fundamentals

Figure 6: Example of ranked information assets

Two additional scoring methods were introduced for comparison against $\mathbf{score}_{\mathbf{u,a}}$. The first (Method 1) is based solely on the structural distance **dists**, representing information assets in the immediate neighbourhood of the current context. The second (Method 2) takes into consideration the proficiency distance **distp** with respect to information assets. The full scoring method $\mathbf{score}_{\mathbf{u,a}}$ (Method 3) adds the collaborative filtering factor **col**.

Scoring method	Method 1 **dists**	Method 2 **dists × distp**	Method 3 **dists × distp × col**
Average MSE	18.0	11.6	9.2

Figure 7: Initial results

The results in Figure 7 indicate that the inclusion of the proficiency distance **distp** in Method 2 made a significant difference to the outcome in comparison to Method 1. However, the full scoring method, Method 3, contributed only marginal improvements thereafter. In hindsight, this was not surprising due to the lack of sufficient usage statistics. Consequently, SME assigned belief values were used and these are unlikely to be agreeable with all users in all topics. The choice and quality of the subjective belief values may be important especially in the initial stages.

Overall, the full ARM scoring strategy clearly contributes toward more accurate recommendations. We expect that, with the availability of reliable usage statistics, the prediction model would progressively become more accurate and reflective of users' actual preferences.

5. Conclusion

In our earlier work [1], we applied knowledge organization strategies and topic map technologies to manage and encourage the construction of dynamically evolving communal knowledge. The Adaptive Recommendation Module (ARM) enhances our earlier work by directing the attention of the user to assets of interest and relevance to him. This helps increase the signal-to-noise ratio in computer-supported collaborative learning environments with a prolific body of evolving knowledge artifacts [3]. Common approaches to this problem include keyword-based clustering and neural networks. In this work, ARM uses topic maps to define the structure and semantic relationships within and between categories and information assets; this addresses the issue of semantic relevance and is specified from the perspective of the subject matter expert. Additionally, the user declares in his user profile the categories of interest to the him and his proficiency level for each. In concert, topic maps and user profiles provide a snapshot of the semantic structures and user-preferences, and are relatively static.

Probabilistic reasoning and Bayesian inference further facilitate collaborative filtering by progressively and dynamically re-evaluating relevance based on the collective experiences of users with similar profiles. ARM incrementally identifies and refines the semantic associations between knowledge artifacts, thus elevating the progressive enquiry process from the personal/private to the collective/collaborative. Initial tests have shown that ARM performs significantly better than recommendations based on semantic structures alone. ARM can be further augmented with peer feedback, influencing user proficiency levels and the degree of contribution to the collaborative filtering process; identification of popular traversal patterns for different user profiles; and organically evolving topic maps influenced by emerging patterns of semantic relationships. In the near-term, we plan to integrate the ARM engine into our production information portal.

References

[1] Looi, C.-K. & Ong, E. (2003). Towards Knowledge Organization in Collaborative Learning Environments. Proceedings of International Conference on Computers in Education, 2003, AACE.

[2] Maurino, A & Fraternali, P (2002). Commercial Tools for the Development of Personalized Web Applications: A Survey. *EC-Web 2002, LNCS 2455*, Bauknecht, Tjoa & Quirchmayr (Eds), pp99-108, 2002.

[3] Ong, E, Looi, C.-K. & Wong, L.-H (2004). From knowledge maps to collaborative interactions. Proceedings of International Conference on Computers in Education, 2004.

[4] Park, J. (2002) Topic Maps, The Semantic Web, and Education. In J. Park (ed), XML Topic Maps – Creating and using topic maps for the Web, Addison Wesley, 2002.

[5] Penuel, B., & Roschelle, J. (1999). Designing learning: Cognitive science principles for the innovative organization. Designing learning: Principles and technologies (SRI paper series). SRI Project 10099.

[6] Pepper, S. (2000). The TAO of Topic Maps. XML Europe 2000. Also available at: http://www.gca.org/papers/xmleurope2000/papers/s11-01.html

[7] Siegel, A, (2000). Towards knowledge organization with Topic Maps, http://www.gca.org/papers/xmleurope2000/papers/s22-02.html.

[8] Wilson, B. & Ryder, M. (1998). Distributed learning communities - an alternative to designed instructional systems, http://carbon.cudenver.edu/~bwilson/dlc.html.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Qualitative and Quantitative Student Models

Jose-Luis Perez-de-la-Cruz [1], Ricardo Conejo and Eduardo Guzmán
Dpt. LCC, ETSI Informática, Universidad de Málaga

Abstract. This paper is a first attempt to relate quantitative, unidimensional models to the fine-grained models usually found in the AI-ED community. More concretely, we define a certain type of qualitative student models that take into account the strict prerequisite relation, and show how a quantitative model arises from it in a natural way.

1. Introduction

In AI-ED literature, we can find proposals to model a student by means of comprehensive, fine-grained structures taking into account, for example, bug libraries, mental models, episodic memory, or learning preferences and styles. These rich, *qualitative* structures are usually difficult to initialize and update for a given student.

The very opposite approach is to model the student by just a real number θ (*performance measure*, in the terminology of [4]). In many real situations (for example, assigning students to groups), students are ranked in function of the results of a test and then the tutorial action is selected. At least as a first approximation, some systems use such an approach, directly or defining fuzzy labels on θ (the system KNOME[1] could be conceptualized in this way). Needless to say, the advantages of such *quantitative* models arise from the existence of well-founded mathematical techniques that allow their easy computation and updating.

A richer model makes feasible a better ITS. However, a more careful consideration shows that this is not always the case [4], [5], [8]. To cite J. Self, "it is not essential that ITSs possess precise student models, containing detailed representations of all the component mentioned above, in order to tutor students satisfactorily" [5]. In fact, "a student model is what enables a system to care about a student" [6], so "there is no practical benefit to be gained from incorporating in our student models features which the tutoring component makes no use of" [5]. On the other hand, it is clear that just a real number will be seldom a powerful model for tutoring; even for assessment tasks, the increasing interest in formative assessment creates the "…challenge of converting each examinee's test response pattern into a *multidimensional* student profile score report detailing the examinee's skills learned and skills needing study" (our emphasis) [7].

[1] Correspondence to: J. L. Perez-de-la-Cruz Dpt. LCC, ETSI Informática, Universidad de Málaga, Bulevar Luis Pasteur s/n, 29071 Málaga, Spain. Tel.: +34 952 132801; Fax: +34 952 131397; E-mail: perez@lcc.uma.es

So a trade-off is needed between the expressive richness of a model and the easiness of its creation and maintenance; and this trade-off will be governed by the gains in "tutoring power" vs. the losses in "creation and updating costs."

The research here presented addresses some of these problems. To this end, we will define a fine-grained structure for modeling student's knowledge and show how a quantitative unidimensional model can be suitably defined from it (section 2). Then we apply this theoretical framework to certain simple cases (section 3) that are amenable to explicit analytical techniques and to more complex cases whose study demands simulation tools (section 4). Finally, the conclusions drawn are summarized and future lines of research are sketched.

2. Theoretical Framework

A domain D is a directed acyclical graph $D(K, A)$ where K —the set of nodes— is the set of *knowledge atoms* and A —the set of arcs— is the *prerequisite relation*, i. e., $k_i \rightarrow k_j$ when the knowledge atom k_j cannot be mastered without mastering the atom k_i. Notice that, in this way, we are considering only *conjunctive* prerequisites. We will denote by N the cardinality of K, i. e., the number of knowledge atoms in the domain.

Given a domain D, a *qualitative student model C* (in the following, a *model*) is a subset of K such that, if $k_i \in C$ and $(k_j, k_i) \in A$, then $k_j \in C$, i. e., a subset of K that satisfies the constraints posed by the prerequisite relation. Notice that we are considering only *binary* valued for the mastering of a knowledge atom, i. e., for each k_i, the student knows totally/does not know the atom.

Let C_1, C_2 be two models. C_1 is a father of C_2 (or, alternatively, C_2 is a son of C_1) when $C_1 \subseteq C_2$ and $card(C_1) = card(C_2) - 1$, i. e., C_1 is a father of C_2 when C_2 can be generated by adding just an atom to C_1 in a way allowed by the prerequisite constraints. We will denote by $\sigma(C)$ the number of sons of C and by $F(C)$ the set of fathers of C.

The weight $w(C)$ of a a model C is defined recursively as follows:

$$w(C) = \begin{cases} 1 & \text{if } C = \emptyset \\ \sum_{C_i \in F(C)} \frac{w(C_i)}{\sigma(C_i)} & \text{otherwise} \end{cases}$$

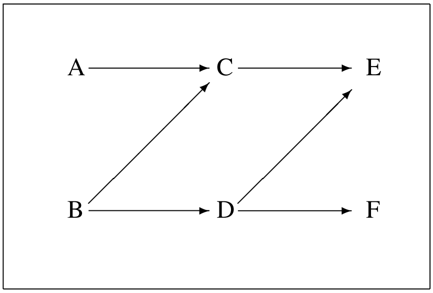
Notice that for each model C, $0 \leq w(C) \leq 1$, and that for each $m, 0 \leq m \leq N$,

$$\sum_{card(C)=m} w(C) = 1.$$

Perhaps an example will clarify the meaning of these definitions. Let us consider the domain of the figure 1(a). There are 6 atoms. Atoms A and B are prerequisites of C; atom B is prerequisite of D; atoms C and D are prerequisites of E; and atom D is prerequisite of F. There are 13 possible models. Their cardinalities and weights are summarized in figure 1(b).

Given a domain D, a *quantitative unidimensional model P* is a real number. It can be termed the student's *knowledge level.*

Now we want to define a function from models into knowledge levels, i. e., a function $f : 2^K \rightarrow \Re$. Some properties are intuitively desirable for the intended function f. For example, given a domain, f must be strictly monotonic, i. e, if $C_1 \subset C_2$, then



C	atoms in C	$card(C)$	$w(C)$
C_1		0	1
C_2	A	1	1/2
C_3	B	1	1/2
C_4	A, B	2	3/4
C_5	B, D	2	1/4
C_6	A, B, C	3	3/8
C_7	A, B, D	3	4/8
C_8	B, D, F	3	1/8
C_9	A, B, C, D	4	11/16
C_{10}	A, B, D, F	4	5/16
C_{11}	A, B, C, D, E	5	11/32
C_{12}	A, B, C, D, F	5	21/32
C_{13}	A, B, C, D, E, F	6	1

(a) (b)

Figure 1. A toy domain (a) and its models (b).

$f(C_1) < f(C_2)$, i. e, if the student knows more atoms, then his knowledge level is greater. The most obvious way is defining f as the count of known atoms $card(C)$, normalized into the common interval $[0, 1]$ and spread along all the real line, for example by means of the antilogistic function:

$$f(C) = \theta_C = log\frac{\frac{card(C)}{N}}{1 - \frac{card(C)}{N}}; \qquad card(C) = n(\theta) = N\frac{1}{1 + e^{-\theta}}$$

Notice that f takes a finite number of values, namely, $N + 1$. When $C = \emptyset$, $\theta_C = -\infty$; when $C = K$, $\theta_C = \infty$.

Let us assume that observable behavior consists of answers to certain questions, called *test items*. The relationship between θ_C and each test item T_i is given by an *Item Characteristic Curve, ICC*, such that $ICC_i(\theta)$ is the probability of giving a right answer to T_i if the student's knowledge is θ. To simplify the exposition, let us assume that every test item T_i depends just on one knowledge atom k_j. Let us also assume that there are neither slips nor guesses, i. e., that a student S answers correctly T_i if and only if $k_j \in C_S$, where C_S is the model corresponding to S's present knowledge. Then $ICC_i(\theta)$ is simply the probability of mastering the knowledge atom k_j given that the knowledge level is θ. The usual expression for an ICC whit no slip nor guess is the logistic function (see, for example, [2])

$$ICC(\theta) = \frac{1}{1 + e^{-a(\theta - b)}}$$

where b is the item difficulty level, such that when $\theta = b$, then $ICC(\theta) = 1/2$; and a is the item discrimination factor, such that when $\theta = b$, $dICC/d\theta = a/4$. Obviously, every $ICC_i(\theta)$ is monotonic.

For our models, a very naive approach would be to define $ICC_i(\theta)$ as follows: (i) count the number $N(\theta)$ of models C whose cardinality is $n(\theta)$; (ii) count the num-

θ	$-\infty$	-1.609	-0.697	0.000	0.693	1.609	∞
ICC_1	0.000	0.500	0.750	0.875	1.000	1.000	1.000
ICC_2	0.000	0.500	1.000	1.000	1.000	1.000	1.000
ICC_3	0.000	0.000	0.000	0.375	0.625	1.000	1.000
ICC_4	0.000	0.000	0.250	0.625	1.000	1.000	1.000
ICC_5	0.000	0.000	0.000	0.000	0.000	0.312	1.000
ICC_6	0.000	0.000	0.000	0.125	0.375	0.687	1.000

Table 1. Values of the ICCs for the domain of figure 1.

ber $N_1(\theta, k_i)$ of models C whose cardinality is $n(\theta)$ and $k_i \in C$; then, $ICC_i(\theta) = N_1(\theta, k_i)/N(\theta)$. However, this definition leads to nonmonotonic functions, i. e., it is possible that $\theta_1 \leq \theta_2$ and $N_1(\theta_1, k_i)/N(\theta_1) > N_1(\theta_2, k_i)/N(\theta_2)$. consider for example a domain with atoms $\{A, B, C, D\}$ and arcs $\{(B, C), (B, D)\}$. There are two models of cardinality 1: $C_1 = \{A\}$ and $C_2 = \{B\}$. $A \in C_1$ but $A \notin C_2$, hence $N_1(\theta_1, A)/N(\theta_1) = 1/2$. However, there are three models of cardinality 2: $C_3 = \{A, B\}$; $C_4 = \{B, C\}$; and $C_5 = \{B, D\}$. $A \in C_3$ but $A \notin C_3$ and $A \notin C_4$. Therefore, $N_1(\theta_2, A)/N(\theta_2) = 1/3$.

In fact, the real definition must take into account the different "likelihood" of every model C. We will adopt the following definition: let Θ_i be the set of models C such that $card(C) = n(\theta)$ and $k_i \in C$. Then $ICC_i(\theta) = \sum_{C \in \Theta_i} w(C)$. In this way, the "likelihood" of a model C is given by the relative number of paths of learning that can lead from the empty state of knowledge to the state represented by C. It is easy to show that $0 \leq ICC_i(\theta) \leq 1$ and that the function so defined is monotonic.

For example, let us show the values of $ICC_i(\theta)$ for the atoms in the domain of figure 1(a). Let us consider atom 1. For $\theta = -\infty$, i. e., $n(\theta) = 0$, there is just a model (the empty one, C_1 in table 1(b)) and $1 \notin C_1$, hence $ICC_1(-\infty) = 0$. For $n(\theta) = 1$, i. e., $\theta = -1.609$, there are two models, C_2 and C_3, with equal weight 1/2. Since $1 \in C_2$ but $1 \notin C_3$, $ICC_1(-1.609) = 0.5$. For $n(\theta) = 2$, i. e., $\theta = -0.697$, there are two models, C_4 and C_5, $w(C_4) = 3/4$, $w(C_5) = 1/4$. Since $1 \in C_4$ but $1 \notin C_5$, $ICC_1(-0.697) = 0.75$. In this way we can compute the values given in table 1.

3. Some Simple Cases

3.1. Lineal Domains

In the simplest cases, it is possible to derive analytically $ICC(\theta)$ and study its relationship to the features of the qualitative underlying model. For example, let us assume that the domain is lineal, i. e., that knowledge atoms are totally ordered,

$k_1 \rightarrow k_2 \rightarrow k_3 \rightarrow \ldots \rightarrow k_p$

In this case, there is exactly one model C_j for each possible cardinality j (therefore, its weight is 1) and $k_i \in C_j$ if and only if $i \leq j$. Therefore,

$$ICC_i(\theta) = \begin{cases} 0 \text{ if } \theta \leq \log \frac{i}{p-i} \\ 1 \text{ otherwise} \end{cases}$$

This is a degenerated logistic function with $a = \infty$ and $b = \log \frac{i}{p-i}$. In other words, the difficulty of k_i is $\log \frac{i}{p-i}$ and its discrimination is ∞. Let us assume now that a test item T_j requires the knowledge of *several* knowledge atoms $k_{j_1}, \ldots, k_{j_m}$. Then ICC_{T_j} is just ICC_{j_m}, i. e., the shape of the function is the same and the parameters are those of the most difficult knowledge atom.

Notice that in such domains given the knowledge level θ, for every knowledge atom k_j we can decide if k_j is known by the student. In this case, if we represent in the model the concrete atoms known by the student there is no gain of information; the quantitative model is an exact representation of the fine-grained one.

3.2. Flat Domains

Let us assume now that the domain is totally flat, i. e., there are no prerequisites. In this case, there are exactly $\binom{N}{j}$ models for each possible cardinality j. Obviously, their weights are equal to $1/\binom{N}{j}$. From these models, $\binom{N-1}{j-1}$ contain a certain atom i. Therefore, all ICCs are the same ICC and

$$ICC(\theta) = \frac{\binom{N-1}{n(\theta)-1}}{\binom{N}{n(\theta)}} = \frac{n(\theta)}{N} = \frac{1}{1+e^{-\theta}}$$

This is a logistic function with $a = 1$ and $b = 0$. In other words, the difficulty of every item is 0 and the discrimination is 1 (or 1/1.7, depending on the normalization adopted). On the other hand, let us assume now that a test item T_j requires the knowledge of several knowledge atoms $k_{j1}, \ldots, k_{jm}$. Analogously we can prove that $ICC(\theta) = \frac{n(\theta)(n(\theta)-1)\ldots(n(\theta)-m+1)}{N(N-1)\ldots(N-m+1)}$ and when $N \to \infty$, $ICC(\theta) \to \frac{1}{(1+e^{-\theta})^m}$. This is not the usual logistic function; however, if we define the difficulty level b as the value of θ such that $ICC(\theta) = 1/2$, then $b = \log \frac{1}{\sqrt[m]{2}-1}$; and, if we define the discrimination factor a as 1/4 times the slope at that point, then $a = m(2 - \sqrt[m]{2})$.

Notice that “the IRT model, in and of itself, simply does not address the question of why some items might be more or less difficult than others” ([3], p. 30); and the same could be asserted about the differences in the discriminating power between different items. However, in flat domains, our approach explain the real nature of these parameters: both difficulty and discrimination are monotone functions of the number m of atoms required to answer the test item. On the other hand, both parameters are assumed independent in IRT theory. If our analysis is correct, it is not the case for flat domains.

4. Some Simulations

For more realistic domains, it becomes impossible to explicitly obtain expressions for response curves. We have developed a simulation tool in order to study empirically the quantitative approximations in those models. With this tool we can define domains structured in levels. Each level contains a number of knowledge atoms. For each atom at a level i, its direct prerequisites are placed at the level $i - 1$. Every atom (for level $i > 1$) has at least one prerequisite.

Different possibilities are allowed by the tool. For example, we can input a given domain with all its nodes and arcs. On the other hand, we can generate a domain at ran-

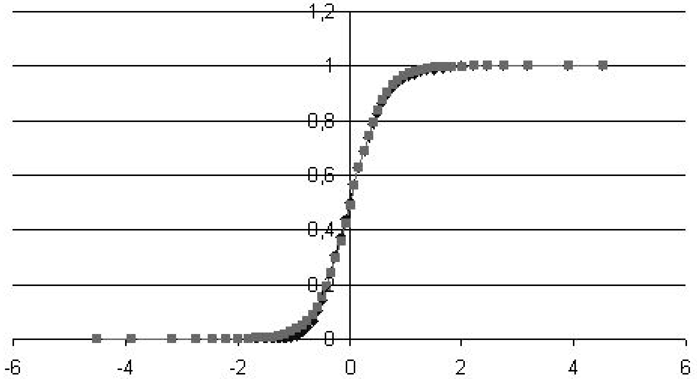


Figure 2. Real and logistic ICC.

dom, giving as input (i) the number of levels; (ii) for each level, the number of atoms; and (3) for each level, the expected number of prerequisites of an atom. In any case, the domain is processed by (i) computing all possible models and their weights; (ii) counting the presence/absence of each atom in each model; (iii) compiling the corresponding ICCs for each knowledge atom. Since the number of domains grows —in general— in an exponential way, this process can be very expensive in space and time. For example, for the domain used to generate the plots shown in this section, there are 50 atoms but 62515 domains (a big number, but distant from 2^{50}, the total number of subsets.) The domain consist of 50 knowledge atoms structured in 5 levels of 10 atoms. The number of prerequisites for each atom is at least 1 and its expected value is 3.

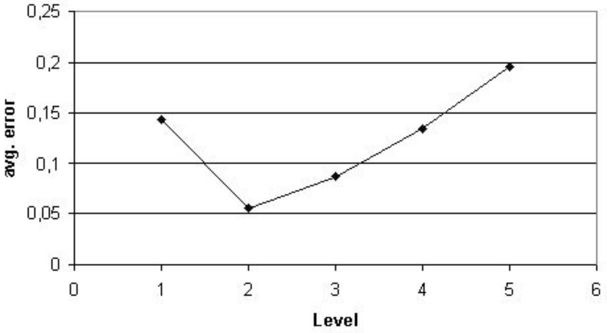


Figure 3. Average error vs. atom level.

The graphics in this section display the relation between some magnitudes in this domain. The aim of the graphics is just showing the kind of problems we are addressing and the kind of answers we are looking for. No claims of generality are made about the hints or tendencies shown by the figures. Not even a statistical analysis of the significance of the data has been performed; in fact, it must wait until a more exhaustive battery of simulations had been performed.

The first issue we want to study is the adequacy of usual logistic ICCs to response curves empirically found. Since we are considering that the response to a test item is deterministically given by the mastery of one knowledge atom, there are 50 response curves, one for each knowledge level. In figure 2 a real ICC is shown and compared to the its best (2 parameter) logistic approximation. The fitness seems good. More formally, the mean value of the quadratic error for the 50 curves is 0,1233.

However, the error is not the same for all atoms. The atom displayed in figure 2 lies "at the middle" of the domain. It can be studied, too, the relation between the level of the atom and the mean error. The results are shown in figure 3. The error is greater for the levels placed at the beginning or at the end of the domain.

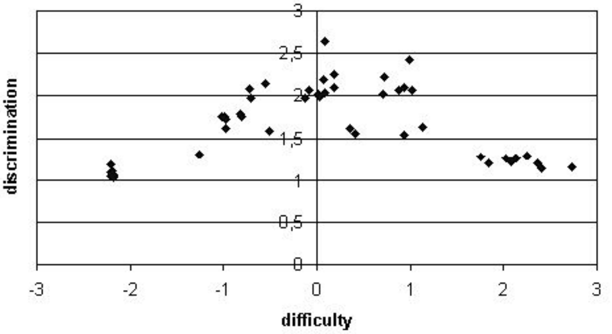


Figure 4. Discrimination vs. difficulty.

Another issue is the study of the correlation between the difficulty and the discrimination of an item. As said in section 3, both parameters are assumed independent. However, figure 4 shows that perhaps it is not the case in real domains.

5. Conclusions and Future Work

We have defined a certain family of qualitative, fine-grained student models. These models, simple as they are, take into account the prerequisite relation. We have derived a quantitative model from the qualitative one and shown how the response curves can be derived. The derivations have been done analytically for some simple cases and by means of simulations in more complex cases.

A lot of work must be done along these lines, with the final aim of determining in which cases quantitative models could be a sensible choice.

References

[1] D. N. Chin, KNOME: Modeling What the user Knows in UC. In Kobsa, A. and Wahlster, W. (eds.): *User Models in Dialog Systems*, Springer, Berlin (1988) 74–107.

[2] S. E. Embretson, S. Reise, *Item Response Theory for Psychologists*. Lawrence Erlbaum, 2000.
[3] Mislevy, R. L.: *Test Theory Reconceived*. CSE Technical Report **376** (1994), National Center for Research on Evaluation, Standards, and Student Testing (CRESST), University of California, Los Angeles.
[4] S. Ohlsson, Some principles of intelligent tutoring. *Instructional Science* **14** (1986) 293–326.
[5] J. Self, Bypassing the Intractable Problem of Student Modelling. In C. Frasson and G. Gauthier (eds.): *Proc. ITS'88*, Ablex, Norwood, N. J. (1988) 107–123.
[6] J. Self, The defining characteristics of intelligent tutoring system research: ITSs care, precisely. *Intl. J. of Artificial Intelligence in Education* **10** (1999) 350–364.
[7] W. Stout, Psychometrics: From Practice to Theory and Back. *Psychometrika* **67** (2002) 485–518.
[8] B. P. Woolf, T. Murray, Using Machine Learning to Advise a Student Model. *Intl. J. of Artificial Intelligence in Education* **3(4)** (1992) 401–416.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Making Learning Design Standards Work with an Ontology of Educational Theories

Valéry PSYCHÉ[1, 2], Jacqueline BOURDEAU[1], Roger NKAMBOU[2], Riichiro MIZOGUCHI[3]
[1] *LICEF, Télé-université, 4750 Henri-Julien, Montréal, (QC) H2T 3E4 Canada*
[2] *GDAC, UQÀM, C.P. 8888, succ. Centre Ville, Montréal, (QC) H3C 3P8 Canada*
[3] *ISIR, Osaka University, 8-1 Mihogaoka, Ibaraki, Osaka, 567-0047 Japan*
{vpsyche, bourdeau}@licef.teluq.uquebec.ca, nkambou.roger@uqam.ca, miz@ei.sanken.osaka-u.ac.jp

Abstract. In this paper, we present an ontology of educational theories their relation to learning design. This ontology takes into account learning design (LD) specifications such as OUNL-EML and IMS-LD at the conceptual level (1), semantic web standards such as OWL at the formal level (2), as well as JAVA standards at the implementation level (3).

This ontology is intended to provide a knowledge base for any IMS-LD compliant authoring systems/LKMS, in order to provide services to authors of LD scenarios. The ontological engineering (OE) has been done using the Hozo ontology editor at levels 1 and 2 respectively.

Introduction

The research presented in this paper follows the initial idea developed in [1] [2] [3], regarding the elicitation through ontological engineering (OE) of instructional design, instruction, learning and knowledge in an authoring system.

The foundations of ontological engineering issues in authoring systems were established in [4] [5], in which we presented (a) a case analysis and (b) the rationale behind it. In (a), specifically, an author assisted by an authoring system or a *Learning and Knowledge Management System* (LKMS) needs to select a relevant learning design (LD) strategy in order to produce a learning scenario. In this case, the author benefits from having access to the theories on which such strategies rely. In (b), we have introduced the rationale for concrete situations in the authoring process that exploit a theory-aware authoring system. In the present article, we propose an ontology of educational theories which describes these theories and their links to the LD, in order to make authoring systems theory-aware. We also discuss the question of having this ontology compliant to e-learning standards in order to provide shareable and reusable services.

Our former research was based on [6] for the representation of the educational theories, and on MISA [7] for that of the learning design process. Recently, in order to enhance and complete these representations, our work has been further inspired by the following: the Open University of the Netherlands' Educational Modeling Language (OUNL-EML) [8] and the IMS Learning Design [9] (IMS-LD) specifications.

In section 1, we give an overview of related work and e-learning technologies standardization efforts. In section 2, we discuss the needs/requirements of authors/learning designers, and the services that an appropriate system could provide in this respect. In section 3, we propose an educational ontology which integrates LD specifications, following which we propose an OWL formalization of this ontology. We conclude in section 4 by summarizing our contribution and by listing our objectives in terms of further work.

1. Overview of Related Work and E-learning Technologies Standardization Efforts

In e-learning, ontologies are increasingly used to organize LD knowledge in authoring systems and LKMS [10] [11] [12] [13] [14]. In most cases, ontologies facilitate the referencing and the retrieval of semantically marked-up learning objects [10] [15]. The most valuable characteristics of ontologies in this respect are shareability, explicitness, and formalism.

Concurrently, recognized standard-initiating organizations have set forth the importance of sharing a common view of the educational field. In 2002, the European Committee for Standardization (CEN/ISSS) conducted a survey of educational modeling languages (EMLs) [16], in which the six existing EMLs were compared. Two distinct groups seemingly emerged. The first, consisting of CDF, LMML, Targeteam, and TML, restricts itself to the modeling of learning content and structure. These languages seem to ignore the existence of pedagogical models. The second consists of PALO and OUNL-EML, and this group lives up to the survey's working definition of EML: "*An EML is a semantic rich information model and binding, describing the content and process within "units of learning" from a pedagogical perspective*" [16]. The survey has shown that the expressive power of OUNL-EML exceeds that of PALO. The OUNL-EML [8] [17], now called EML, aims at providing a pedagogical meta-model. It consists of four extendable models which describe: (a) how learners learn (based on a consensus among learning theories); (b) how units of studies which are applicable in real practice are modeled, given the learning model and the instruction model; (c) the type of content and the organization of that content; and (d) the theories, principles and models of instruction as they are described in the literature or as they are conceived in the mind of practitioners. EML and its subsequent integration to IMS-LD has been to date the most important initiative towards integrating instructional design preoccupations in the international e-learning standardization effort [7].

IMS-LD [9] takes the EML information model as its base. For binding purposes, it is made compatible with the IMS specifications: CP, QTI, CD, SS [9]. The LD is positioned as the containment framework for all these specifications allowing instructional design (called "Learning Design" in IMS and henceforth in this paper) to be included into content packages. According to [9] "*A Learning Design is a description of a method enabling learners to attain certain learning objectives by performing certain learning activities in a certain order in the context of a certain learning environment. A learning design is based on the pedagogical principles of the designer and on specific domain and context variables*". In this definition, the place of educational theories in the LD specification is not clear. As a result, however, it underlines the importance of educational theories in the LD specification, since most of existing LD tools fail to explicitly integrate educational theories.

Indeed, the current learning technologies standards and specifications mainly focus on describing knowledge about learning design and content (e.g. LOM, Dublin Core, SCORM, CANCORE), thus offering only limited support to describe knowledge of the educational theories. Consequently, authors/learning designers cannot rely on assistance stemming from theories in their learning design process. Why are LD standards so limited? It may be because of the lack of representation of this theoretical knowledge as well as the lack of a compliance mechanism between these standards and this theoretical knowledge. Such a problem has been one of the concerns of the Learning Object Repository Network (LORNET) research network in Canada. LORNET is developing an authoring environment in the form of a LKMS compliant with IMS-LD standards; we believe that such an LKMS could benefit from providing authors with access to LD theories in order to enhance the quality of their design, and to improve their expertise. "*A taxonomy of pedagogies is a common request as this would enable people to search for learning designs according to the embedded pedagogy*" [17]. In order to thus make LD standards work with a representation of LD theories, a technical solution is needed.

2. Why linking LD Standards to a Representation of Educational Theories?

Assuming that the main user is an author/learning designer, this section introduces: the needs of an author for such a knowledge representation, the resulting services he/she can expect from an appropriate system, and how theses services can be supported through the binding of LD standards to theories. Our goal is consequently to provide services whose specific purpose would be linked to consultation of theories, eventually linking such theories to learning designs based on those theories.

Some needs of the author using an authoring system, as suggested in [5] [18], are the following: (a) Query about which theories apply best to a specific LD, or about design principles related to theories; (b) Extract, (re)view and browse among theories in order to select LD strategies, or among templates of LD scenarios; (c) Review examples of good LD scenarios or principles in order to design a LD scenario; (d) Reuse or modify a template of LD scenario; (e) Validate (check consistency) among design principles.

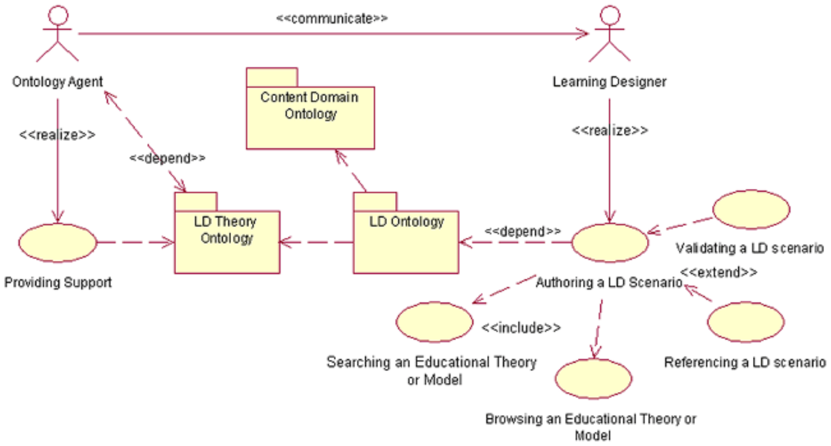


Figure 1. Main Use Cases and Provided Services

Table 1. Example of a Service: Searching a Theory

Use Case Goal	**Perform a search to find a suitable theory**	
Success End Condition	The suitable theory is found and provided by an agent.	
Failed End Condition	No input from the author or no matching theory.	
Primary Actor	**User:** Author / Learning Designer	
DESCRIPTION	**Step**	**Branching Action**
The author searches for appropriate theories for sequencing instruction that would map an LD activity structure in a particular LD scenario.	1	**Author:** wants to select a given type of instructional activity
	2	**Ontology Agent:** consults the ontology
	3	**Ontology Agent:** performs queries as to which theories could map the learning design activity
	4	**Ontology Agent:** outputs a list of suitable theories from the ontology
	5	**Author** selects a theory item in the list

Such a system should therefore assist an author in designing scenarios while improving expertise gained in LD. More specifically, this system should provide the following services [12]: (a) Assist the author in the selection of an appropriate LD method with regards to a scenario and encourage the application of a wide range of available LD methods when requested; (b) Inform this author about a particular LD method when queried; (c) Check and highlight errors in the authoring/design of a scenario when validation is needed/required. (d) Provide relevant examples. These services can be provided through a repository of LD scenarios [17] linked to a learning design ontology, as illustrated in Fig. 1. The LD ontology itself consequently depends on the LD theory ontology and the content domain ontology (cf. section 3 for details). Fig. 1 also shows that searching, browsing, referencing and validation services are common requests. Some of these could be directly provided by a software agent to the author (searching, browsing), while other services (referencing, validation), could be provided through an authoring system or LKMS. Table 1 shows a detailed use case of a search that might be conducted by an author indicating the type of support potentially given by the agent.

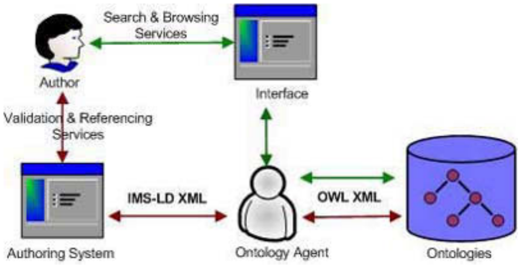


Figure 2. Interactions between Agent and Author during Authoring Process

Fig. 2 shows the interactions and flow of information between the agent and the author while providing those services. The possibility of using LD standards for other services [19] [20], is also explored. For instance, in the case of a validation service, an agent aware of the LD standards would be able to highlight errors or check the consistency of a scenario during the authoring process. This means that with a representation of the LD, an agent would be able to follow and assist the author in the process of authoring a LD scenario. This active assistance is possible only if the ontologies involved are well formalized given that the agent will need to query and reason about the elements within the ontology, which also explains why OWL is used (cf. section 4 for details). Clearly, an author would benefit from these services if the LD was linked to a representation of theories. We assume that the authoring system or LKMS used for indirect services is compliant with LD standards. On the basis of these hypotheses, we now propose how LD standards and LD theories could be connected.

3. Integration of LD Standards through Representation and Binding with Theories

This section describes the solution that has been developed in order to realize this integration: 1) an EML representation in the ontology, 2) a binding mechanism between LD and theories. As a preliminary to this discussion, we first elaborate on our OE methodology:

3.1. Methodology

Our methodology follows the three main steps of OE (before implementation): 1) analysis , 2) conceptualization, 3) formalization, followed by an evaluation [21] and documentation of the ontology.

- *Analysis of the domain.* This step was done by creating a glossary of terms, and includes the following tasks: (a) Identifying each the type of each term (Class, Properties, Individuals); (b) Adding an informal description for each term; (c) Adding synonyms and acronyms if available;
- *Conceptualization.* The conceptual modeling includes the following tasks:
(a) Creating models of classes; (b) Creating *ad hoc* property models.
- *Formalization.* This step was conducted using Hozo [5]. For each class: (a) Add the subclasses in order to create taxonomies of classes; (b) Add predefined properties; (c) Add *ad hoc* properties; (d) Add comments (or annotations) if necessary; (e) Add axioms if

necessary. This is an iterative process, which stops once the ontology is stabilized. Finally; (f) Add individuals.

- *Evaluation*. This step [21] is performed during the conceptualization and formalization steps: (a) Verification: check (assisted by the editor) if the ontology is syntactically correct. (b)Validation: make sure (with domain experts) that the ontology correctly models the real world (domain) for which it was created.
- *Documentation.* At this stage, we document the ontology using OWL terminology: (a) *Creating a dictionary of classes*. For each class, indicate the: identifier, equivalent class, super and sub-classes, individuals, class property; (b) *Creating a dictionary of properties*. For each property, indicate the: name, type, domain, range, characteristics, restrictions; (c) *Creating a dictionary of class axioms*: indicate boolean combinations; (d) *Creating a dictionary of individuals*. For each individual, indicate the: individual name, type name, ObjectPropertyValue, DataPropertyValue.

3.2. An Ontological Conceptualization Compliant with EML & IMSL-LD

We argued previously that LD standards have a very limited connection to theories. Because IMS-LD [9] relies upon EML, we examined the EML meta-model [8] and how LD relates to theories in this meta-model. Fig. 3 shows that the “Unit of Study” is at its heart and relates to theories, to content domain and to learning models. In our view, ontologies could try to match this structure and we thus propose a structure consisting of three ontologies (Fig. 4), in which the “Learning Design Ontology” corresponds to the “Unit of study” and includes the “Learning Model”, while relating to the two other ontologies, the “Learning Design Theories”, and the “Content Domain” Ontology.

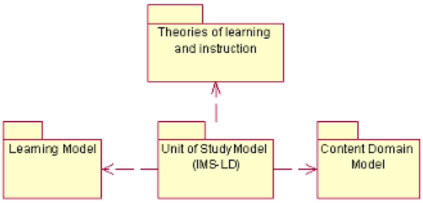

Theories of learning and instruction
Learning Model
Unit of Study Model (IMS-LD)
Content Domain Model

Figure 3. The EML meta-model

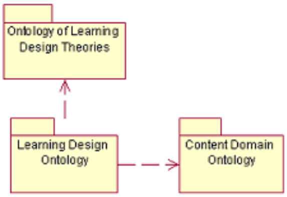
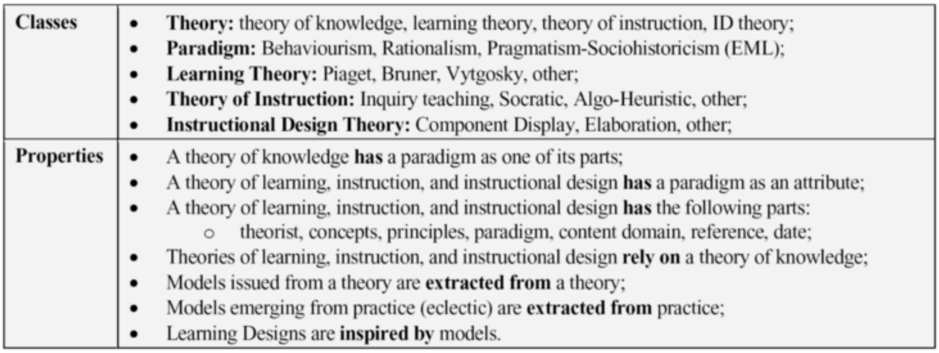


Figure 4. The resulting ontologies

This conceptualization builds upon the ontology of theories presented in [4], and takes into account the classes proposed by EML [8] and extracted from [22]. Classes for theories in EML are paradigm-based: “behaviourism”, “rationalism”, and “pragmatism-sociohistoricism”.

Table 2. Classes and Properties of the Ontology of Educational Theories



Classes	• **Theory:** theory of knowledge, learning theory, theory of instruction, ID theory; • **Paradigm:** Behaviourism, Rationalism, Pragmatism-Sociohistoricism (EML); • **Learning Theory:** Piaget, Bruner, Vytgosky, other; • **Theory of Instruction:** Inquiry teaching, Socratic, Algo-Heuristic, other; • **Instructional Design Theory:** Component Display, Elaboration, other;
Properties	• A theory of knowledge **has** a paradigm as one of its parts; • A theory of learning, instruction, and instructional design **has** a paradigm as an attribute; • A theory of learning, instruction, and instructional design **has** the following parts: o theorist, concepts, principles, paradigm, content domain, reference, date; • Theories of learning, instruction, and instructional design **rely on** a theory of knowledge; • Models issued from a theory are **extracted from** a theory; • Models emerging from practice (eclectic) are **extracted from** practice; • Learning Designs are **inspired by** models.

It appears that these classes correspond, in our ontology, both to the theory of knowledge on which each theory of learning relies, and to the main paradigms identified, although the names sometimes differ [23] [24] [25]. Although these classes should allow for classifying all theories of learning, instruction and instructional design, EML adds another class, called "eclectic", for learning design models that have emerged from practice as opposed to being based on theory. This "other" class has therefore been added to our ontology. Table 2 shows the classes and properties which consequently were obtained as a result of the conceptualization. As a result Fig. 5 shows an UML representation of the theories which binds with the IMS-LD. The main entities of the ontology (theory, paradigm, model, domain and LD) are in grey.

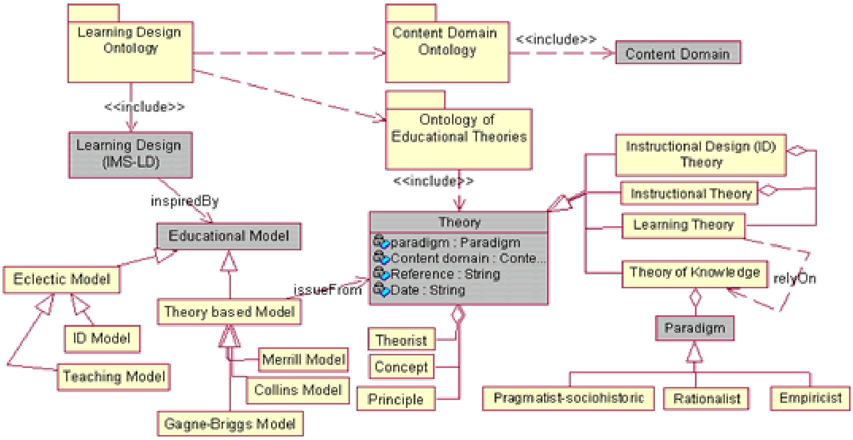


Figure 5. A UML representation of the ontology of theories

What theories are mapped to the LD, and how? Table 2 illustrates examples of how we conceive the binding mechanism between LD and educational theories.

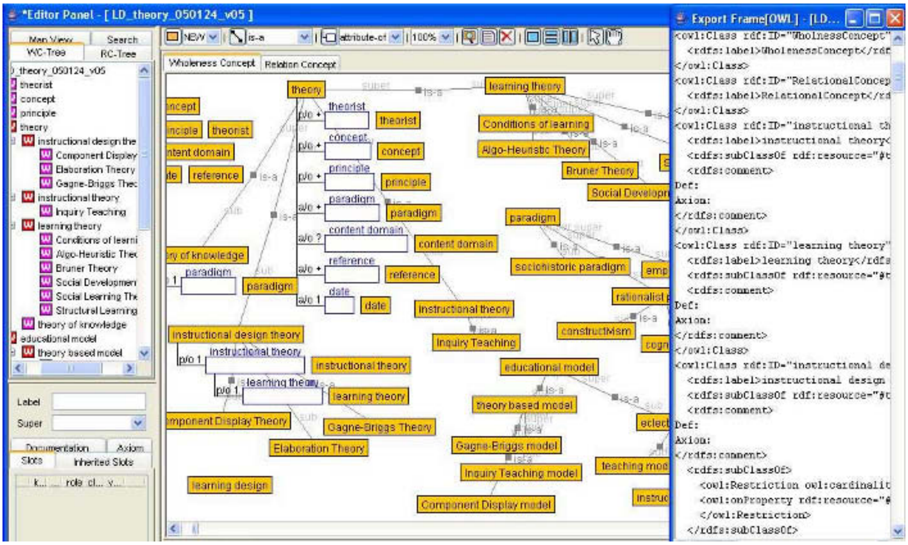
Table 3. An Excerpt of the Binding Mechanism

IMS-LD Element	Binding by Properties	Matching Classes (C) /Instances (I) of Theory
Method	**Type of Paradigm**: * Instructivist (Behaviourist) * Constructivist (Rationalist) * Socioconstructivist (Sociohistoric)	(C): * Gagné Th., Merrill Th., ... * Piaget's Th., Collins' Th., Bruner Th., …. * Vygotsky's Th., Wenger's Th., ...
Learning Objective	**Type of Learning:**	(C): Mager's Th., Bloom's Th., Gagné's Th., … (I): Reigeluth's learning objectives [6]
Support / Learning Activity	**Control of Learning:** * Teacher-centered * Learner-centered * Team-based	(C) * Gardner's Th., Gagné's Th., Merrill's Th., … * Piaget Th, Collins Th., Bruner Th., … * Vygotsky's Th., Wenger Th., ...
Activity Structure	**Sequencing of Instruction**	(C) Gagné-Briggs' events, Collin's techniques...

4. Formalizing and Implementing the Ontology for Agent Use

The software agent receives a LD scenario description and retrieves a selection of matching theories available on a web-based knowledge base using a set of emerging standards (RFD-S, OWL) and tools (Hozo, Jena2). To achieve this goal, a formalization (level 2 in [26]) followed by an implementation (level 3 in [26]) of the ontology was necessary.

The formalization was done in OWL (*Web Ontology Language*) using the Hozo ontology editor. OWL is designed for use by applications that need to process information in addition to displaying information to humans. In comparison to XML, RDF, and RDF Schema (RDF-S), it facilitates better machine interpretability of Web content since it provides additional vocabulary along with a formal semantics. OWL has three increasingly-expressive sublanguages: OWL Lite, OWL DL, and OWL Full [27]. Our formalization was conducted using OWL DL. The Hozo editor allows for the creation of classes and properties, in addition to a graphic representation of the ontology, the hierarchy of classes and the properties. It also generates the OWL code as shown in Fig. 6 (right window).



```
<owl:Class rdf:ID="WholnessConcept"
  <rdfs:label>WholenessConcept</rdf
</owl:Class>
<owl:Class rdf:ID="RelationalConcep
  <rdfs:label>RelationalConcept</rd
</owl:Class>
<owl:Class rdf:ID="instructional th
  <rdfs:label>instructional theory<
  <rdfs:subClassOf rdf:resource="#t
  <rdfs:comment>
Def:
Axiom:
</rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="learning theory"
  <rdfs:label>learning theory</rdfs
  <rdfs:subClassOf rdf:resource="#t
  <rdfs:comment>
Def:
Axiom:
</rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="instructional de
  <rdfs:label>instructional design
  <rdfs:subClassOf rdf:resource="#t
  <rdfs:comment>
Def:
Axiom:
</rdfs:comment>
  <rdfs:subClassOf>
    <owl:Restriction owl:cardinalit
    <owl:onProperty rdf:resource="#
    </owl:Restriction>
  </rdfs:subClassOf>
```

Figure 6. Formalization of the Ontology of Educational Theories in Hozo

A subsequent ontology implementation using Jena2 is in progress. Jena2, developed by Hewlett-Packard, is a Java framework for programming Semantic Web applications. It includes useful features, including an ontology API, a reasoning system, a query language (RDQL). The *ontology API* offers support for the implementation of the above-formalized ontologies (RDFS, OWL) into JAVA classes. The *reasoning system*, an inference engine, together with rule sets for RDFS / OWL, works with the ontology API in order to infer additional facts from a particular ontology source. *RDQL* offers support for querying a networked knowledge base consisting of the above elements, and allows the agent to query the ontology of theories about elements of LD scenario specified by the author.

5. Conclusion

In merging LD standards with an ontology of theories to serve the needs of authors working within an authoring system or a LKMS, we found that IMS-LD cannot link the learning design with instructional design theories. We developed a solution that integrates LD in a structure of ontologies, and allows for communication between LD and theories. We described the ontology of theories with its classes and properties. A first version has been formalized in OWL using the Hozo ontology editor. This work needs to be further developed to provide the services expected by its users. The ontology also needs to be merged with the ontology of the three instructional models (Gagne-Briggs, Merrill and Collins) that has been previously developed [4]. Furthermore, a deeper integration of LD standards is envisaged within an ontology of LD. The agent will be implemented according to JAVA standards. At this point, our work will be interfaced with the LKMS developed by LORNET. Both an evaluation of the ontology and of the services provided by the agent are foreseen. The evaluation of the ontology itself then follows criteria and guidelines by [21]. The services provided by the agent to a learning designer in the process of authoring using an IMS-LD compliant tool will be evaluated in the following way: a mockup will represent the interactions between the agent and the human author, in the context of a real task. Three LD experts will judge the services' relevance, usefulness and meaningfulness.

Acknowledgments

Thanks to Ophélie Tremblay, Michel Leonard, Karin Lundgren, Ioan Rosca, Olga Marino and Gilbert Paquette (LICEF), and to Danièle Allard (Mizoguchi Lab.), for their advice or feedback. Financial support was provided by the Canadian National Science and Engineering Research Council, the LICEF Research Center and GDAC Research Lab.

References

[1] Bourdeau J. and Mizoguchi R., "Collaborative Ontological Engineering of Instructional Design Knowledge for an ITS Authoring Environment," *ITS*, pp. 399-409, 2002.

[2] Mizoguchi R. and Bourdeau J., "Using Ontological Engineering to Overcome Common AI-ED Problems," *Int. Journal of AI in Education*, vol. 11, pp. 107-121, 2000.

[3] Mizoguchi R., Sinitsa K., and Ikeda M., "Knowledge engineering of educational systems for authoring systems design," *Euro AIED*, pp. 329-335, 1996.

[4] Psyché V., Mizoguchi R., and Bourdeau J., "Ontology Development at the Conceptual Level for Theory-Aware ITS Authoring Systems.," *AIED*, pp. 491-493, 2003.

[5] Bourdeau J., Mizoguchi R., Psyché V., and Nkambou R., "Potential of an Ontology-based ITS Authoring Environment: One Example," *ITS*, pp. 150-161, 2004.

[6] Reigeluth C. M., "Instructional Theories in Action," LEA, 1993, pp. 343.

[7] Paquette G., *Instructional Engineering for Network-based Learning*: Wiley-Pfeiffer, 2003.

[8] Koper R., "Modeling Units of Study from a Pedagogical Perspective," 2001.

[9] IMS Global Learning Consortium, "IMS LD, CP, QTI, CD and SS Specifications," http://www.imsglobal.org/specificationdownload.cfm. Last consulted, April 2005.

[10] Recker M. and Wiley D., "A non-authoritative Educational Metadata Ontology for Filtering and Recommending Learning Objects," *Journal of Interactive Learn. Environ.*, vol. 9, pp. 255-271, 2001.

[11] Psyché V., Mendes O., and Bourdeau J., "Apport de l'ingénierie ontologique aux environnements de formation à distance," in *STICEF*, vol. 10, Hotte R. and Leroux P., Eds.: STICEF, 2003, pp. 89-126.

[12] Meisel H. and al., "An Ontology-Based Approach to Intelligent Instructional Design Support," *KES*, 2003.

[13] Amorim R. and al., "An Educational Ontology based on Metadata Standards," *ECEL*, pp. 29-36, 2003.

[14] Aroyo L., Inaba A., Soldatova L., and Mizoguchi R., "EASE," *ITS*, pp. 140-149, 2004.

[15] Leidig T., "L3 Towards an Open Learning Envir.," *ACM Journal of Edu. Res. in Comp.*, vol. 1, pp. 7, 2001.

[16] Rawlings A., Rosmalen van P., Koper R., Artacho M., and Lefrere P., "Survey of Educational Modelling Languages (EMLs)," CEN/ISSS WS/LT 2002.

[17] Koper R. and Olivier B., "Representing the Learning Design of Units of Learning," *Educational Technology & Society*, vol. 7, pp. 97-111, 2004.

[18] Nkambou R., Frasson C., and Gauthier G., "Authoring Tool for Knowledge Engineering in ITS," in *Authoring Tools for Advanced Technology Learning Env.*, Murray T. and al., Eds., 2003, pp. 93-138.

[19] Psyché V., "CIAO, an Interface Agent Prototype to facilitate the use of ontology in intelligent authoring system," *Annual Scientific Conference of the LORNET Research Network*, 2004.

[20] van Rosmalen P., Boticario J., and Santos O., "The Full Life Cycle of Adaptation in aLFanet eLearning Environment," *IEEE Computer Society LTTC*, vol. 6, pp. 4, 2004.

[21] Gomez-Perez A., "Ontology Evaluation," in *Handbook on Ontologies*, Staab and Studer, Eds., 2003.

[22] Greeno J., Collins A., and Resnick L., "Cognition and Learning," *Handbook of Educational Psychology*, pp. 15-46, 1996.

[23] Ertmer P. and Newby T., "Behaviorism, cognitivism, constructivism," vol. 6, pp. 50-70., 1993.

[24] Mayer R. E., "Learners as information processors," *Educational Psychologist*, vol. 31, pp. 151-161, 1996.

[25] Kearsley G., "Explorations in Learning & Instruction: The Theory Into Practice Database," 1994-2004.

[26] Mizoguchi R., "A Step Towards Ontological Engineering," *12th Conf. on AI of JSAI*, pp. 24-31, 1998.

[27] W3C Consortium, "OWL Specification Development," http://www.w3.org/2004/OWL/#specs Feb 2004.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Detecting the Learner's Motivational States in An Interactive Learning Environment

Lei Qu and W. Lewis Johnson
USC / ISI, 4676 Admiralty Way, Suite 1001, Marina del Rey, CA, 90292
{leiqu, johnson}@isi.edu

Abstract. It is important for pedagogical agents to have the ability to detect the learner's motivational states. With this ability, agents will be more sensitive to the cognitive and emotional states of the learner and be able to promote the learner's motivation through interaction with the learner. In this paper we present a method for agents to assess learner's motivational states in an interactive learning environment. It takes into account the learner's attention, current task and expected time to perform the task. An experiment was conducted to collect data for evaluating the performance of the method, and the results showed that there is more than 75% to detect the learner's motivational states where intervention is warranted.

Introduction

In Intelligent Tutoring System (ITS), it is extremely important for pedagogical agents to be able to influence the learner's affective state. To support this in ITSs, agents must be able to recognize the learner's affective state and understand how it changes. The major assumption is that, with information on the learner's affective state, agents can interact more socially with the learner.

Other researchers have proposed methods for recognizing the learner's affective states. Conati uses biometric sensors to monitor the leaner's emotions in educational games [1]. Picard described some models of affective and motivational states (e.g. interest, stuck and frustration [10][11]), using special sensors (e.g. head tracker, pressure mouse and chair with a posture sensor). De Vicente [12] described a model to detect various motivational states (e.g. interest, effort, satisfaction) based on the learner's performance and activities such as mouse movement, quality and speed of performance. However the detection model was based on insufficient knowledge on learner's task and focus attention. This insufficiency frequently results in inaccurate detecting.

This work aims at enabling pedagogical agents to assess the learner's motivational states in an analogical way to what a human tutor does in an interactive learning environment. It untilizes knowledge on learner's task and focus of attention without requiring any special device other than an ordinary video camera. In our work, we modeled the learner's motivational states (confidence, confusion and effort), and performed an experimental study to evaluate our method. This paper is organized as follows: Section 1 introduces the background studies; Section 2 describes the motivational model; Section 3 describes the experimental study; Section 4 summarizes our evaluation results for this model; and Section 5 is a discussion about future work.

1. Background

In an earlier study, we investigated how human tutors coach learners while interacting with the Virtual Factory Teaching System (VFTS), which is an on-line factory system for teaching engineering concepts and skills [2]. We conducted follow-on studies in which a tutor assisted learners via a chat-based interface. From these studies we noted that the tutors were making assessments about the learners’ affective and motivational states, and using these state assessments to decide when and how to assist the learners. There are many states that can be used by the tutor to assess the learner’s motivation. Researchers in motivation such as Harackiewicz [3] and Lepper et al. [4] have identified many states, such as curosity, confidence and control. Some of the most important learner’s states in our studies were confidence, confusion, and effort as defined in Table 1.

Table 1. Definition for motivational states

State	Definition
Confidence	This reflects the learner’s confidence of solving problems in the learning environment.
Confusion	This defines the degree of hesitancy while the learner makes decision.
Effort	This measures the duration of time that the learner spends on performing tasks.

It was found that human tutors frequently use the following types of information to infer the learner’s motivation:

- The learner’s task/goal
- The learner’s focus of attention
- The frequency of the learner’s questions

Therefore the work discussed in this paper aims at investigating whether the three motivational states in Table 1 can be automatically inferred based on these infomation. To this end, we design a new system with the user interface shown in Figure 1, and two models to enable an agent to have access to the information listed above.

The new interface includes three major components:

- The VFTS interface, which reports each keyboard entry and mouse click that the learner performs on it.
- WebTutor, which is an on-line tutorial that explains how to employ the VFTS to perform common industrial engineering tasks (forcasting product demand, planning manufacturing steps, and scheduling the manufacturing jobs).
- The Agent Window, in which the left part is a text window used to communicate with the agent (or a human tutor in Wizard-of-Oz mode) and the right part is an animated character that is able to generate speech and gestures.

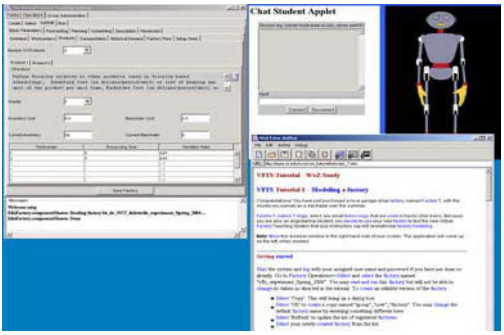

Chat Student Applet
VFTS Tutorial - WoZ Study
VFTS Tutorial 1 - Modeling a factory

Figure 1. User interface

Meanwhile the new system includes two additional models to track the learner's attention and activities:

- The attention tracking model [5] is used to infer the learner's focus of attention. It uses a Bayesian model to combine the information from the eye gaze program (developed by Larry Kite at the Laboratory for Computational and Biological Vision at USC) and interface events. The eye gaze program estimates the coordinates on a video display that correspond to the focus of gaze in order to track the learner's eye focus. The tracking model then informs agents which window is the focused window of the learner: VFTS, Webtutor Window, Agent Window, or other area.
- The plan recognizer [5] is used to track the learner's progress in the VFTS. It identifies what current plan of the learner is likely to be, based upon what learner is reading in the tutorial, and then tracks the learner as he/she performs each step in the plan. For each task in the plan, plan recognizer has an estimate of how much time is required by a typical learner to read the paragraph, decide what action to take, and carry out that action. The information from the plan recognizer includes six variables as listed in Table 2.

The input devices consist of keyboard, mouse, and a camera focused on the learner's face. This interface thus provides information that is similar to the information that human tutors use in tracking the learner activities. A Wizard-of-Oz study was then conducted with the interface to verify that the information collected via the interface was sufficient for agents to track the learner's activities.

Table 2. Definitions of information from plan recognizer

Variable	Definition
EstActionTime	Estimated time to perform the task.
EstReadTime	Estimated time to read the paragraph related to this task.
EstDecisionTime	Estimated time for the learner to decide how to perform the task.
StartTime/EndTime	The time when the learner starts/finishes a task.
Progress	The number of tasks that learner has finished with respect to the current plan.
ErrorTries	The number of unexpected tasks performed by the learner which are not included in current plan.

2. Modelling Learner's Motivational States

This section describes how the learner's motivational states are modelled in our system.

2.1 Modelling Confidence

There are three major sources of information for a human tutor to infer learner's confidence: 1) the learner's hesitancy in performing actions after reading the tutorial; 2) the history of task performance (for example, how many tasks the learner has successfully completed.); and 3) the frequency of the learner's requests for help on certain tasks. For example, if the learners perform actions in the VFTS after reading tutorial without much hesitancy, this implies that they must have high confidence. Following the above empirical observations, we therefore model the learner's confidence focusing on the following aspects:

- *Progress* in completing the current plan, reported by the plan recognizer.
- *ErrorTries* for the current plan, as reported by the plan recognizer.
- The number of questions that the learner types in the Agent window to request for help.
- *StartTime* and *EstActionTime* for the current plan, reported by the plan recognizer.

The learner's confidence is then modelled as one of three levels: *High*, *Normal* and *Low*. *Normal* is the initialized default level. In order to dynamically measure the learner's confidence during the tasks, three inference rules are employed:

- If the learner has made progress (i.e. the value of *Progress* is increased) within the duration of *EstActionTime* for the current plan, then the learner's confidence will be increased by one level (e.g. from *Low* to *Normal*, or from *Normal* to *High*).
- If the learner has used up the *EstActionTime* and made some error tries (i.e. the value of *ErrorTries* is increased), but failed to make any progress (i.e. the value of *Progress* is not increased), then learner's confidence will be decreased by one level (e.g. from *Normal* to *Low*, or from *High* to *Normal*).

If the learner has asked any question about the current plan (i.e. the number of questions is increased), then the learner's confidence will be decreased by one level.

2.2 Modelling Confusion

Another important motivational state is confusion, which reflects the learner's failing to understand the tutorial or deicide how to proceed in the VFTS. A learner with high confusion is most likely to be stuck or frustrated. The following factors are considered by agents to infer the learner's level of confusion.

- *Progress* and *ErrorTries* for the current plan, from the plan recognizer.
- *EstReadTime*, *EstDecisionTime* and *EstActionTime* for the step.
- The number of the learner's questions, as discussed in Section 2.1.
- The learner's reading time t_{read}, decision time $t_{decision}$ and action time t_{action} for current step. These three variables are the actual time that the learner spends in system. The t_{read} and $t_{decision}$ are obtained from attention tracking model. The t_{action} is obtained from the plan recognizer.

With information provided by the above factors, the agents can derive the learner's confusion level to be one of the three levels: *High*, *Normal*, or *Low* by the following four inference rules:

- If the learner has made some progress (i.e. *Progress* is increased) during the duration of read, decision and action time (i.e. $t_{read} + t_{decision} + t_{action}$), then confusion will be decreased by one level (e.g. from *High* to *Normal*, or from *Normal* to *Low*).
- If the learner has not made any progress or any error try (i.e. *Progress* and *ErrorTries* remain unchanged) during the duration of read, decision and action time (i.e. the sum of t_{read}, $t_{decision}$, and t_{action}), then confusion will be increased by one level (e.g. from *Normal* to *High*, or from *Low* to *Normal*).
- If the learner has made some error tries without any progress (i.e. *ErrorTries* is increased but *Progress* remains unchanged) during the duration of read, decision and action time, then confusion will be increased by one level.

- If the learner has asked any question about the current plan, then the learner's confusion will be increased by one level.

Both confusion and confidence are used to measure the learner's degree of indecision. However, they provide agents with different insights on choosing different strategies to intervene with learner. For example, confusion is primarily used for agent to detect learner's confusion or frustration. For the learner with high confusion, an agent tutor should give more explicit instruction. But for learner with low confidence, agent should motivate the learner by a socratic hint (or polite suggestion, [6]).

2.3 Modelling Effort

By estimating how much time the learner has already spent on a task, a human tutor can infer the learner's effort for this task. Based on how the human tutor infers the learner's effort during in-person interactions, we derive the inference rules for agents to detect the learner's effort.

The formula used to measure the effort value (*EV*) relating to a certain task is $EV = t_s/t_{e,}$ where t_s is the period of time that the learner has already spent on fulfilling a certain task, and t_e represents the estimated time/duration that is needed for the learner to complete this task. The t_s includes the learner's t_{read}, $t_{decision}$ and t_{action} inferred from the attention tracking model and the plan recongizer as discussed in Section 2.1. The time duration of t_e includes the estimated reading time *EstReadTime*, decision time *EstDecisionTime*, and action time *EstActionTim* from the plan recognizer.

For the learner's current plan $plan_m$ with n tasks, $task_i$ ($i=1, 2...n$), we can get EV_i ($i=1, 2...n$) related to the relative effort the learner spends on fulfilling $task_i$ in the VFTS. So the agent uses the average value of EV_i ($i=1, 2...n$) to evaluate how much effort the learner devotes to $plan_m$. If the learner does not complete all the tasks of $plan_m$, then agents will take the average effort's value of only those completed tasks. Such value is therefore considered as the learner's effort value (*EV*) for a task or a plan. If the learner already completes several plans in VFTS, agents will calculate the average *EV* for all these plans as the learner's current *EV*.

We define two threshold values for the learner's effort: $threshold_{low}$ and $threshold_{high}$ as 0.8 and 1.0 respectively. Learner's effort levels (*EL*) are determined as *High*, *Normal* or *Low* based on learner's t_{read}, $t_{decision}$, t_{action} and *EV*.
by following the three rules:

- $EL = High$: when $EV > threshold_{high}$, $t_{read} > 0$, $t_{decision} > 0$ and $t_{action} > 0$
- $EL = Normal$: when ($EV <= threshold_{high}$ and $EV > threshold_{low}$) or any one of t_{read}, $t_{decision}$ and t_{action} equals to 0
- $EL = Low$: when $EV <= threshold_{low}$ or any two of t_{read}, $t_{decision}$ and t_{action} equal to 0.

3. Experimental Study

3.1 Method

To evaluate our method, we designed and conducted an experimental study to collect the data. With new interfaces and models, we ran 24 subjects at the University of California at Santa Barbara. The 24 participants were all undergraduate students. Most of them had computer skills but little or no knowledge of industrial engineering.

In the experiment, each participant read a tutorial in the Webtutor to learn some concepts in industrial engineering and how to work in the VFTS, and then performed actions in the VFTS to carry out the tasks described in the tutorial. A human tutor observed the learner's activities using Wizard-of-Oz interface [7] in another room. When the tutor felt the learner was having difficulties, she intervened and provided appropriate help. Also the learner could request help by clicking the "Request help" button in the Agent Window.

3.2 Data Collection and Pre-processing

The learner's motivation can be assessed accroding to various data collection approaches, for example, by direct observations, ratings by human tutors, or self-reports by the learner [8]. In our experiment, the collected data were classified into the following four datasets:

- Dataset A, which is from the screen capture of the learner's interface,
- Dataset B, which consisted of the learner interface data such as keyboard events and mouse events,
- Dataset C, which is from a self-report completed by the learner at the end of each phase. After the learner completes a phase, the human tutor sent the learner an on-line questionnaire to report his/her motivational states (confidence, confusion and effort). In this self-report, the learner reported his/her motivational states by a three-level-scale: *High*, *Normal*, and *Low*. The system saved the data into a database with a timestamp. Human tutor only sent the learner a self-report questionnaire after he/she finished one phase in order to avoid disturbing the learner's work and thereby undermining the learner's motivation, and
- Dataset D, which was the learner's inferred focus of attention and task progress as determined by the attention tracking model and the plan recognizer.

After the experiment, the dataset A and B were imported into anvil [9], a video annotation tool that supports annotation of video with multi-layer information. The human tutor that interacted with the learner in the experiment then watched the recorded data in anvil, inferred the learner's motivational states (confidence, confusion and effort) and reported their value as *High*, *Normal* and *Low*. Such data was saved with a timestamp as dataset E, which was used as the basis for accuracy evaluation of our model and will be discussed in Section 4.

4. Results

4.1 Evaluation

24 runs were performed with durations ranging from 30 minutes to 70 minutes. The average time that the learners spent on with the system was around 40 minutes. Based on Dataset D (the learner's attention and activities information from the attention tracking model and the plan recognizer), our model infers the learner's motivational states at the scales of *High*, *Normal*, and *Low* at every second. The inferred data from our model includes timestamp, state and level.

Our model was evaluated using two methods with respect to two different comparison datasets. Method I compared the inferred data from the model with that in Dataset E. Dataset E is the learner's motivation as inferred by human tutor. As shown in Table 3, the human tutor has recorded 351 datapoints about the learners' motivation based on his/her observations after the review of the experiment data. Each datapoint included a motivational state (e.g. confidence, confusion or effort), its corresponding level (e.g. *High*, *Normal* or *Low*), a timestamp, and a comment about the level. Method II compared the inferred data with that in Dataset C (the learner's motivation from self-report). There were 123 datapoints in dataset C with the same format reported by the learners as discussed earlier in Section 3.

Table 3. Summary for Dataset E and C

	Total	Confidence	Confusion	Effort
Dataset E	351	78	138	135
Dataset C	123	41	41	41
Total	474	119	179	176

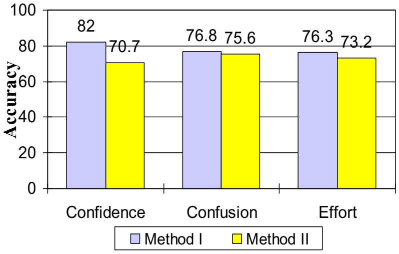


Figure 2. Evaluation results for the accuracy of our model

The results of these two evaluations are shown in Figure 2. It can be seen that the recognition accuracy is 82.0%, 76.8% and 76.3% for the learner's confidence, confusion and effort when Dataset E is comparison set. And the recongnition accuracy dropped to 70.7%, 75.6% and 73.2% when Dataset C is comparison set. As expected, the model has a higher recognition for the learner's motivation when we use Dataset E as comparison set. The drop of recognition accuracy for confidence may be caused by the inconsistent judgement for different learners.

Furthermore, certain situations for these motivational states are considered particularly important because they situate where an agent tutor should be proactive in assisting or influencing the learner. These include situations when: 1) learner confidence is low, 2) learner confusion is high, and 3) learner effort is low. To further investigate on these, Figure 3(a) defines four categories of evaluation results for comparing our model prediction with a comparison set. The "true positive" (TP) cases are instances where the model predicted the tutor's or learner's assessment that the target condition exists; "false positive" (FP) instances are cases where the model indicated that the target condition exists but the tutor disagreed; "true negative" (TN) cases are instances where the model predicted the tutor's assessment that the target condition does not exist and "false negative" (FN) cases are instances where the model indicated that the target condition exists but the tutor disagreed. Figure 3(b) shows the evaluation results of "true positive", "false positive", "true negative" and "false negative" cases compared with Dataset E (human tutor's judgement) and Dataset C (learner's self-report). For example, 32.1% out of 42.4% (the sum of 32.1% and 10.3% for confidence) positive conditions can be recongized by our method, this makes the accuay of 75.7%.

		Model Prediction	
		Negative	Positive
Comparison Set Assessment	False	FN	FP
	True	TN	TP

(a) Evaluation Matrix

Motivational States	Recognition rate (%)							
	Dataset E				Dataset C			
	TP	FP	TN	FN	TP	FP	TN	FN
Confidence	32.1	10.3	30.8	10.2	14.6	9.7	17.1	12.2
Confusion	45.7	15.2	18.1	3.6	19.5	7.3	26.8	7.3
Effort	18.5	5.9	44.4	14.1	17.1	9.7	19.5	7.3

(b) Recognition rate for TP, FP, TN and FN in Dataset E and C

Figure 3. Evaluation results for three selected situations

5. Conclusion

Learner's attention and activities information are important for inferring the learner's motivation. We have used such information to construct a motivational model to infer learner's motivational factors in an interactive learning environment. Such model can infer learner's motivation at any given moment.

In conlcusion, we can say that the results of our evaluation suggest that such model can provide agents accurate information about learner's motivation. It is possible for pedagogical agents to detect learner's motivation with confidence and provide learner with proactive help in order to motivate the learer's learning.

6. ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation under Grant No. 0121330, and in part by a grant from Microsoft Research. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation. We thank members of CARTE group who contributed to this work, especially Ning Wang for tutoring in experimental studies and reviewing the data.

References

[1] Conati, C., Chabbal, R., and Maclaren, H., A Study on Using Biometric Sensors for Detecting User Emotions in Educational Games. In: Proceedings of the Workshop "Assessing and Adapting to User Attitude and Affects: Why, When and How? ". In conjunction with UM '03, 9th International Conference on User Modeling, Pittsburgh, PA, 2003.

[2] Johnson, W. L., Interaction Tactics for Socially Intelligent Pedagogical Agents. In Proceedings of IUI '03, International Conference on Intelligent User Interfaces, Miami, Florida, 2003, pp. 251-253.

[3] Sansone, C., and Harackiewicz, J. M., Intrinsic and extrinsic motivation: The search for optimal motivation and performance. San Diego: Academic Press, 2000.

[4] Lepper, M. R., Woolverton, M., Mumme, D., and Gurtner, J., Motivational techniques of expert human tutors: Lessons for the design of computer-based tutors. In S.P. Lajoie and S.J. Derry (Eds.), Computers as cognitive tools, Hillsdale, NJ: Lawrence Erlbaum Associates, 1993, pp. 75-105.

[5] Qu, L., Wang, N., and Johnson, W. L., Choosing when to interact with learners. Intelligent User Interfaces 2004, pp. 307-309.

[6] Johnson, W. L., Rizzo, P., Bosma, W., Kole, S., Ghijsen, M., and Welbergen, H. V., Generating Socially Appropriate Tutorial Dialog. ADS 2004, pp. 254-264.

[7] Johnson, W. L., Rizzo, P., Lee, H., Wang, N., and Shaw, E., Modeling Motivational and Social Aspects of Tutorial Dialog. ITS Workshop on Human Tutorial Tactics and Strategies, 2005.

[8] Pintrich, P. R., and Schunk, D. H., Motivation in Education: Theory, Research, and Applications 2nd, Merill Precentice Hall, 1995.

[9] Kipp, M., Anvil-A Generic Annotation Tool for Multimodal Dialogue. In Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech), Aalborg, 2001, pp. 1367-1370.

[10] Burleson, W., and Picard, R. W., Affective Agents: Sustaining Motivation to Learn Through Failure and a State of Stuck, Social and Emotional Intelligence in Learning Environments Workshop In conjunction with the 7th International Conference on Intelligent Tutoring Systems, Maceio - Alagoas, Brasil, August 31st, 2004.

[11] Dennerlein, J., Becker, T., Johnson, P., Reynolds, C., and Picard, R. W., Frustrating Computer Users Increases Exposure to Physical Factors, Proceedings of the International Ergonomics Association, Seoul, Korea, August 24-29, 2003.

[12] De Vicente, A., and Pain, H., Informing the detection of the students' motivational state: An empirical study. In S.A. Cerri, G. Gouardères, & F. Paraguaçu (Eds.), Intelligent Tutoring Systems: 6th International Conference, ITS 2002, pp. 933-943.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Blending Assessment and Instructional Assisting

Leena RAZZAQ[*], Mingyu FENG, Goss NUZZO-JONES, Neil T. HEFFERNAN, Kenneth KOEDINGER[+], Brian JUNKER[+], Steven RITTER[°], Andrea KNIGHT[+], Edwin MERCADO*, Terrence E. TURNER, Ruta UPALEKAR, Jason A. WALONOSKI, Michael A. MACASEK, Christopher ANISZCZYK, Sanket CHOKSEY, Tom LIVAK, Kai RASMUSSEN

Department of Computer Science
Worcester Polytechnic Institute, Worcester, MA, USA
assistments@wpi.edu

[+]*Human-Computer Interaction Institute*
Carnegie Mellon University, Pittsburgh, PA, USA

[°]*Carnegie Learning, Inc.*

Abstract. Middle school mathematics teachers are often forced to choose between assisting students' development and assessing students' abilities because of limited classroom time available. To help teachers make better use of their time, we are integrating assistance and assessment by utilizing a web-based system ("Assistment") that will offer instruction to students while providing a more detailed evaluation of their abilities to the teacher than is possible under current approaches. An initial version of the Assistment system was created and used last May with about 200 students and 800 students are using it this year once every two weeks. The hypothesis is that Assistments both assist students while also assessing them. This paper describes the Assistment system and some preliminary results.

Introduction

Limited classroom time available in middle school mathematics classes compel teachers to choose between time spent assisting students' development and time spent assessing students' abilities. To help resolve this dilemma, assistance and assessment are integrated in a web-based system ("Assistment"[1]) that will offer instruction to students while providing a more detailed evaluation of their abilities to the teacher than is possible under current approaches. The plan is for students to work on the Assistment website for about 20 minutes per week. The Assistment system is an Artificial Intelligence program. Each week when students work on the website, the system "learns" more about the students' abilities and thus, it can hypothetically provide increasingly accurate predictions of how they will do

[*] This research was made possible by the US Dept of Education, Institute of Education Science, "Effective Mathematics Education Research" program grant #R305K03140, the Office of Naval Research grant # N00014-03-1-0221, NSF CAREER award to Neil Heffernan, and the Spencer Foundation. Authors Razzaq and Mercado were funded by the National Science Foundation under Grant No. 0231773. All the opinions in this article are those of the authors, and not those of any of the funders.

[1] The term “Assistment” was coined by Kenneth Koedinger and blends Assessment and Assisting.

on a standardized mathematics test. The Assistment System is being built to identify the difficulties individual students – and the class as a whole – are having. It is intended that teachers will be able to use this detailed feedback to tailor their instruction to focus on the particular difficulties identified by the system. Unlike other assessment systems, the Assistment technology also provides students with intelligent tutoring assistance while the assessment information is being collected.

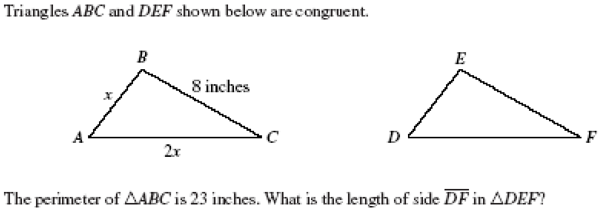
An initial version of the Assistment was created and tested last May. That version of the system included 40 Assistment items. There are now approximately 150 Assistment items. The key feature of Assistments is that they provide instructional assistance in the process of assessing students. The hypothesis is that Assistments can do a better job of assessing student knowledge limitations than practice tests or other on-line testing approaches by using a "dynamic assessment" approach. In particular, Assistments use the amount and nature of the assistance that students receive as way to judge the extent of student knowledge limitations. Initial first year efforts to test this hypothesis of improved prediction of the Assistment's dynamic assessment approach are discussed below.

In preparation for fall of 2004, 75 Assistment items were created and 9 teachers and about 1000 students are currently using them in 3 schools. Currently, there are approximately 150 Assistments.

1. Assistment System and website development

In December of 2003, one of the authors met with the Superintendent of the Worcester Public Schools in Massachusetts, and was subsequently introduced to the three math department heads of 3 out of 4 Worcester middle schools. The goal was to get these teachers involved in the design process of the Assistment System at an early stage. The main activity done with these teachers was meeting about one hour a week to do "knowledge elicitation" interviews, whereby the teachers helped design the pedagogical content of the Assistment System.

The procedure for knowledge elicitation interviews went as follows. A teacher was shown a Massachusetts Comprehensive Assessment System (MCAS) test item and asked how she would tutor a student in solving the problem. What kinds of questions would she ask the student? What hints would she give? What kinds of errors did she expect and what would she say when a student made an expected error? These interviews were videotaped and the interviewer took the videotape and filled out an "Assistment design form" from the knowledge gleaned from the teacher. The Assistment was then implemented using the design form. The first draft of the Assistment was shown to the teacher to get her opinion and she was asked to edit it. Review sessions with the teachers were also videotaped and the design form revised as needed. When the teacher was satisfied, the Assistment was released for use by students.

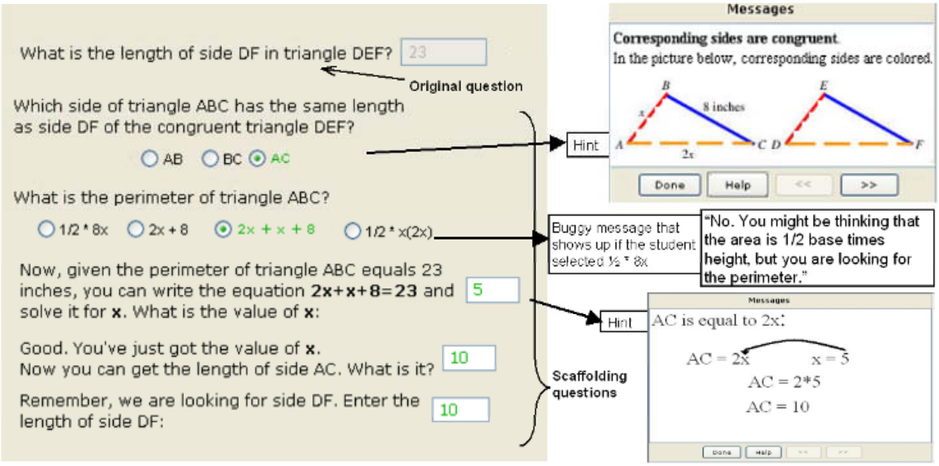

Triangles ABC and DEF shown below are congruent.

The perimeter of $\triangle ABC$ is 23 inches. What is the length of side $\overline{DF}$ in $\triangle DEF$?

Figure 1: Item 19 from the 2003 MCAS

For instance, a teacher was shown a MCAS item on which her students did poorly, such as item #19 from the year 2003, which is shown in Figure 1. About 15 hours of knowledge elicitation interviews were used to help guide the design of Assistments.

Figure 2 shows an Assistment that was built for the item 19 shown above. Each Assistment consists of an *original item* and a list of *scaffolding questions* (in this case, 5 scaffolding questions). The first scaffolding question appears only if the student gets the item wrong. Figure 2 shows that the student typed "23" (which happened to be the most common wrong answer for this item from the data collected). After an error, students are not allowed to try the item further, but instead must then answer a sequence of scaffolding questions (or "scaffolds") presented one at a time[2]. Students work though the scaffolding questions, possibly with hints, until they eventually get the problem correct. If the student presses the hint button while on the first scaffold, the first hint is displayed, which is the definition of congruence in this example. If the student hits the hint button again, the hint that is shown in Figure 2 appears, which describes how to apply congruence to this problem. If the student asks for another hint, the answer is given. Once the student gets the first scaffolding question correct (by typing AC), the second scaffolding question appears.



What is the length of side DF in triangle DEF? 23

Which side of triangle ABC has the same length as side DF of the congruent triangle DEF?

○ AB ○ BC ● AC

What is the perimeter of triangle ABC?

○ 1/2 * 8x ○ 2x + 8 ● 2x + x + 8 ○ 1/2 * x(2x)

Now, given the perimeter of triangle ABC equals 23 inches, you can write the equation **2x+x+8=23** and solve it for **x**. What is the value of **x**: 5

Good. You've just got the value of **x**. Now you can get the length of side AC. What is it? 10

Remember, we are looking for side DF. Enter the length of side DF: 10

Messages

Corresponding sides are congruent

In the picture below, corresponding sides are colored.

Buggy message that shows up if the student selected ½ * 8x

"No. You might be thinking that the area is 1/2 base times height, but you are looking for the perimeter."

Messages

AC is equal to 2x:

AC = 2x x = 5

AC = 2*5

AC = 10

Figure 2: An Assistment shown just before the student hits the "done" bottom, showing two different hints and one buggy message that can occur at different points.

If the student selects ½ * 8x, the *buggy message* shown would appear suggesting that it is not necessary to calculate area. (*Hints* appear on demand, while *buggy messages* are responses to a particular student error). Once the student gets the second question correct, the third appears, and so on. Figure 2 shows the state of the interface when the student is done with the problem as well as a hint for the 4th scaffolding question.

About 200 students used the system in May 2004 in three different schools from about 13 different classrooms. The average length of time was one class period per student. The teachers seemed to think highly of the system and, in particular, liked that real MCAS items were used and that students received instructional assistance in the form of scaffolding questions. Teachers also like that they can get online reports on students'

[2] As future work, once a predictive model has been built and is able to reliably detect students trying to "game the system" (e.g., just clicking on answer) students may be allowed to re-try a question if they do not seem to be "gaming". Thus, studious students may be given more flexibility.

progress from the Assistment web site and can even do so while students are using the Assistment System in their classrooms. The system has separate reports to answer the following questions about **items**, **student, skills** and student **actions**: Which **items** are my students finding difficult? Which **items** are my students doing worse on compared to the state average? Which **students** are 1) doing the best, 2) spending the most time, 3) asking for the most hints etc.? Which of the approximately 80 **skills** that we are tracking are students doing the best/worst on? What are the exact **actions** that a given student took?

The three teachers from this first use of the Assistment System were impressed enough to request that all the teachers in their schools be able to use the system the following year. Currently that means that about 1,000 students are using the system for about 20 minutes per week for the 2004-2005 school year. Two schools have been using the Assistment System since September. A key feature of the strategy for both teacher recruitment and training is to get teachers involved early in helping design Assistments through knowledge elicitation and feedback on items that are used by their students.

Assistments are based on Intelligent Tutoring System technology that is deployed with an internet-savvy solution (for more technical details on the runtime see [6]). In the first year's solution, when students started an Assistment item, a Java Web Start application was downloaded and reported each students' actions (other than their mouse movements) to a database at WPI, thus enabling completely live database reporting to teachers. Database reporting for the Assistment Project is covered extensively in [3]. In the second year, the application has been delivered via the web and requires no installation or maintenance. We have spent considerable time observing its use in classrooms; for instance, one of the authors has logged over 50 days, and was present at over 300 classroom periods. This time is used to work with teachers to try to improve content and to work with students to note any misunderstandings they sometimes bring to the items. For instance, if it is noted that several students are making similar errors that were not anticipated, the "Assistment Builder" [4] web-based application can be logged into and a buggy message added that addresses the students' misconception. The application is being prepared for its statewide release in May 2005.

The current Assistment System web site is at www.assistment.org, which can be explored for more examples.

2. Analysis of data to determine whether the system reliably predicts MCAS performance

One objective the project had was to analyze data to determine whether and how the Assistment System can predict students' MCAS performance. In Bryant, Brown and Campione [2], they compared traditional testing paradigms against a dynamic testing paradigm. In the dynamic testing paradigm a student would be presented with an item and when the student appeared to not be making progress, would be given a prewritten hint. If the student was still not making progress, another prewritten hint was presented and the process was repeated. In this study they wanted to predict learning gains between pretest and posttest. They found that static testing was not as well correlated (R = 0.45) as with their "dynamic testing" (R = 0.60).

Given the short use of the system in May, there was an opportunity to make a first pass at collecting such data. The goal was to evaluate how well on-line use of the Assistment System, in this case for only about 45 minutes, could predict students' scores on a 10-item post-test of selected MCAS items. There were 39 students who had taken the posttest. The paper and pencil posttest correlated the most with MCAS scores with an R-value of 0.75.

A number of different metrics were compared for measuring student knowledge during Assistment use. The key contrast of interest is between a static metric that mimics paper practice tests by scoring students as either correct or incorrect on each item, with a dynamic assessment metric that measures the amount of assistance students need before they get an item correct. MCAS scores for 64 of the students who had log files in the system were available. In this data set, the static measure does correlate with the MCAS, with an R-value of 0.71 and the dynamic assistance measure correlates with an R-value of -0.6. Thus, there is some preliminary evidence that the Assistment System may predict student performance on paper-based MCAS items.

It is suspected that a better job of predicting MCAS scores could be done if students could be encouraged to take the system seriously and reduce "gaming behavior". One way to reduce gaming is to detect it [1] and then to notify the teacher's reporting session with evidence that the teacher can use to approach the student. It is assumed that teacher intervention will lead to reduced gaming behavior, and thereby more accurate assessment, and higher learning.

The project team has also been exploring metrics that make more specific use of the coding of items and scaffolding questions into knowledge components that indicate the concept or skill needed to perform the item or scaffold correctly. So far, this coding process has found to be challenging, for instance, one early attempt showed low inter-rater reliability. Better and more efficient ways to use student data to help in the coding process are being sought out. It is believed that as more data is collected on a greater variety of Assistment items, with explicit item difficulty designs embedded, more data-driven coding of Assistments into knowledge components will be possible.

Tracking student learning over time is of interest, and assessment of students using the Assistment system was examined. Given that there are approximately 650 students using the system, with each student coming to the computer lab about 7 times, there was a table with 4550 rows, one row for each student for each day, with an average percent correct which itself is averaged over about 15 MCAS items done on a given day. In Figure 3, average student performance is plotted versus time. The y-axis is the average percent correct on the original item (student performance on the scaffolding questions is ignored in this analysis) in a given class. The x-axis represents time, where data is bunched together into month, so some students who came to the lab twice in a month will have their numbers averaged. The fact that most of the class trajectories are generally rising suggests that most classes are learning between months.

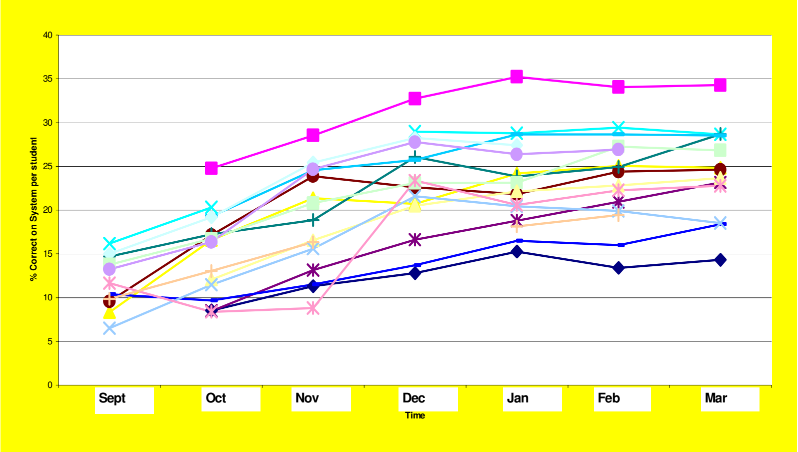


Figure 3: Average student performance is plotted versus time.

Given that this is the first year of the Assistment project, new content is created each month, which introduces a potential confounder of item difficulty. It could be that some very hard items were selected to give to students in September, and students are not really learning but are being tested on easier items. Next year, this confound will be eliminated by sampling items randomly. Adding automated applied longitudinal data analysis [7] is currently being pursued.

3. Analysis of data to determine whether the system effectively teaches.

The second form of data comes from within Assistment use. Students potentially saw 33 different problem pairs in random order. Each pair of Assistments included one based on an original MCAS item and a second "morph" intended to have different surface features, like different numbers, and the same deep features or knowledge requirements, like approximating square roots. Learning was assessed by comparing students' performance the first time they were given one of a pair with their performance when they were given the second of a pair. If students tend to perform better on the second of the pair, it indicates that they may have learned from the instructional assistance provided by the first of the pair.

To see that learning happened and generalized across students and items, both a student level analysis and an item level analysis were done. The hypothesis was that students were learning on pairs or triplets of items that tapped similar skills. The pairs or triplet of items that were chosen had been completed by at least 20 students.

For the student level analysis there were 742 students that fit the criteria to compare how students did on the first opportunity versus the second opportunity on a similar skill. A gain score per item was calculated for each student by subtracting the students' score (0 if they got the item wrong on their first attempt, and 1 if they got it correct) on their 1st opportunities from their scores on the 2nd opportunities. Then an average gain score for all of the sets of similar skills that they participated in was calculated. A student analysis was done on learning opportunity pairs seen on the same day by a student and the t-test showed statistically significant learning ($p = 0.0244$). It should be noted that there may be a selection effect in this experiment in that better students are more likely to do more problems in a day and therefore more likely to contribute to this analysis.

An item analysis was also done. There were 33 different sets of skills that met the criteria for this analysis. The 5 sets of skills that involved the most students were: Approximating Square Roots (6.8% gain), Pythagorean Theorem (3.03% gain), Supplementary Angles and Traversals of Parallel Lines (1.5% gain), Perimeter and Area (4.3% gain) and Probability (3.5% gain). A t-test was done to see if the average gain scores per item were significantly different than zero, and the result ($p = 0.3$) was not significant. However, it was noticed that there was a large number of negative average gains for items that had fewer students so the average gain scores were weighted by the number of students, and the t-test was redone. A statistically significant result ($p = 0.04$) suggested that learning should generalize across problems. The average gain score over all of the learning opportunity pairs is approximately 2%. These results should be interpreted with some caution as some of the learning opportunity pairs included items that had tutoring that may have been less effective. In fact, a few of the pairs had no scaffolding at all but just hints.

4. Experiments

The Assistment System allows randomized controlled experiments to be carried out. At present, there is control for the number of items presented to a student, but soon the system will be able to control for time, as well. Next, two different uses of this ability are described.

4.1 Do different scaffolding strategies affect learning?

The first experiment was designed as a simple test to compare two different tutoring strategies when dealing with proportional reasoning problems like item 26 from the 2003 MCAS: "The ratio of boys to girls in Meg's chorus is 3 to 4. If there are 20 girls in her chorus, how many boys are there?" One of the conditions of the experiment involved a student solving two problems like this with scaffolding that first coached them to set up a proportion. The second strategy coached students through the problem but did not use the formal notation of a proportion. The experimental design included two items to test transfer. The two types of analyses the project is interested in fully automating is to 1) to run the appropriate ANOVA to see if there is a difference in performance on the transfer items by condition, and 2) to look for learning during the condition, and see if there is a disproportionate amount of learning by condition.

Two types of analyses were done. First, an analysis was done to see if there was learning during the conditions. 1st and 2nd opportunity was treated as a repeated measure and to look for a disproportionate rate of learning due to condition (SetupRatio vs. NoSetup). A main effect of learning between first and second opportunity ($p = 0.05$) overall was found, but the effect of condition was not statistically significant ($p = 0.34$). This might be due to the fact that the analysis also tries to predict the first opportunity when there is no reason to believe those should differ due to controlling condition assignment. Given that the data seems to suggest that the SetupRatio items showed learning a second analysis was done where a gain score (2nd opportunity minus 1st opportunity) was calculated for each student in the SetupRatio condition, and then a t-test was done to see if the gains were significantly different from zero and they were ($t = 2.5$, $p = 0.02$), but there was no such effect for NoSetup.

The second analysis done was to predict each student's average performance on the two transfer items, but the ANOVA found that even though the SetupRatio students had an average score of 40% vs. 30%, this was not a statistically significant effect.

In conclusion, evidence was found that these two different scaffolding strategies seem to have different rates of learning. However, the fact that setting up a proportion seems better is not the point. The point is that it is a future goal for the Assistment web site to do this sort of analysis automatically for teachers. If teachers think they have a better way to scaffold some content, the web site should send them an email as soon as it is known if their method is better or not. If it is, that method should be adopted as part of a "gold" standard.

4.2 Are scaffolding questions useful compared to just hints on the original question?

An experiment was set up where students were given 11 probability items. In the first condition, the computer broke each item down into 2-4 steps (or scaffolds) if a student got the original item wrong. In the other condition, if a student made an error they just got hints upon demand. The number of items was controlled for. When students completed all 11 items, they saw a few items that were morphs to test if they could do “close”-transfer problems.

The results of the statistical analysis were showing a large gain for those students that got the scaffolding questions, but it was discovered that there was a selection-bias. There were about 20% less students in the scaffolding condition that finished the curriculum, and those students that finished were probably the better students, thus invalidating the results. This selection bias was possible due to a peculiarity of the system that presents a list of assignments to students. The students are asked to do the assignments in order, but many students choose not to, thus introducing this bias. This will be easy to correct by forcing students to finish a curriculum once they have started it. New results are expected inside a month.

Conclusion

The Assistment System was launched and presently has 3 middle schools using the system with all of their 8th grade students. Some initial evidence was collected that the online system might do a better job of predicting student knowledge because items can be broken down into finer grained knowledge components. Promising evidence was also found that students were learning during their use of the Assistment System. In the near future, the Assistment project team is planning to release the system statewide in Massachusetts.

References

[1] Baker, R.S., Corbett, A.T., Koedinger, K.R. (2004) Detecting Student Misuse of Intelligent Tutoring Systems. *Proceedings of the 7th International Conference on Intelligent Tutoring Systems*, 531-540.

[2] Campione, J.C., Brown, A.L., & Bryant, N.R. (1985). Individual differences in learning and memory. *In R.J. Sternberg (Ed.). Human abilities: An information-processing approach,* 103-126. New York: W.H. Freeman.

[3] Feng, M., Heffernan, N.T., (2005). Informing Teachers Live about Student Learning: Reporting in the Assistment System. *Submitted to the Workshop on Usage Analysis in Learning Systems at 12th Annual Conference on Artificial Intelligence in Education 2005*, Amsterdam.

[4] Turner, T. E., Macasek, M. A., Nuzzo-Jones, G., Heffernan, N.T., (2005). The Assistment Builder: A Rapid Develoment Tool for ITS. *Submitted to the 12th Annual Conference on Artificial Intelligence in Education 2005,* Amsterdam

[5] Koedinger, K. R., Aleven, V., Heffernan. T., McLaren, B. & Hockenberry, M. (2004) Opening the Door to Non-Programmers: Authoring Intelligent Tutor Behavior by Demonstration. *Proceedings of 7th Annual Intelligent Tutoring Systems Conference*, Maceio, Brazil. page162-173

[6] Nuzzo-Jones, G., Walonoski, J. A., Heffernan, N.T., Livak, T.(2005). The eXtensible Tutor Architecture: A New Foundation for ITS. *Submitted to the 12th Annual Conference on Artificial Intelligence in Education 2005*, Amsterdam

[7] Singer, J. D. & Willett, J. B. (2003). Applied Longitudinal Data Analysis: Modeling Change and Occurrence. Oxford University Press, New York.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

A First Evaluation of the Instructional Value of Negotiable Problem Solving Goals on the Exploratory Learning Continuum

Carolyn ROSÉ, Vincent ALEVEN, Regan CAREY, & Allen ROBINSON
Human-Computer Interaction Institute and Department of Mechanical Engineering
Carnegie Mellon University,
5000 Forbes Avenue, Pittsburgh PA 15213

Abstract. We evaluate the effectiveness of a tutorial-dialogue based approach to guided exploratory learning involving problem solving goals that are negotiated between tutor and student rather than dictated by the tutor or freely chosen by the student. This approach, referred to as Negotiable Problem Solving Goals (NPSG), is located on a previously untested space on what we call The Exploratory Learning Continuum. The results of our empirical classroom investigation provide design recommendations for a new type of tutorial dialogue system and strong evidence in favor of tutorial dialogue support in exploratory learning environments.

Introduction

The tutorial dialogue literature provides us with many convincing proofs of the technical feasibility of tutorial dialogue systems [e.g., 7,13,6,1]. What is needed now is insight on how to weild that technology to benefit student learning beyond what is possible with more standard forms of interaction supported by state-of-the-art tutoring systems. Looking at naturalistic human tutorial dialogue inspires us to broaden our view of what intelligent tutoring systems can provide to students, and to consider forms of interaction that are not typically supported by current intelligent tutoring systems. One of the major research goals of the CycleTalk project [14] has been to investigate the instructional effectiveness of novel ways of using tutorial dialogue technology in an exploratory learning environment.

We investigate two separate dimensions that have framed much of the literature on exploratory learning. We evaluate the effectiveness of a tutorial-dialogue based approach involving problem solving goals negotiated between tutor and student rather than dictated by the tutor or freely chosen by the student. This approach, which we refer to as Negotiable Problem Solving Goals, is located on a previously untested space on what we call The Exploratory Learning Continuum. Although our experimental manipulation involves the use of human tutors, the results of our investigation provide design recomendations for a new type of tutorial dialogue system that holds promise for demonstrating the potential contribution tutorial dialogue technology can make to the field of Intelligent Tutoring. In the remainder of the paper we review the exploratory learning literature, the specifics of our experimental design, an analysis of our results, and development plans.

1. Review of Background Literature on Exploratory Learning

A popular conceptualization of exploratory learning is that what distinguishes "exploratory learning" from "non-exploratory learning" is the level at which goals are provided to the learner. Exploratory learning is associated with "high level goals" such as "survive in this simulation environment" or under specified goals such as "find all implications that can be drawn from these premises". In contrast, non-exploratory learning is associated with "low level goals" or "fully specified goals" such as "solve this equation" or "verify whether this implication is true". Non-exploratory learning in this conceptualization may involve means-ends analysis; however the search is directed down a small number of "correct" paths. We argue that all learning is exploratory, and alternative learning tasks or learning environments can be placed along a continuum, which we refer to as The Exploratory Learning Continuum.

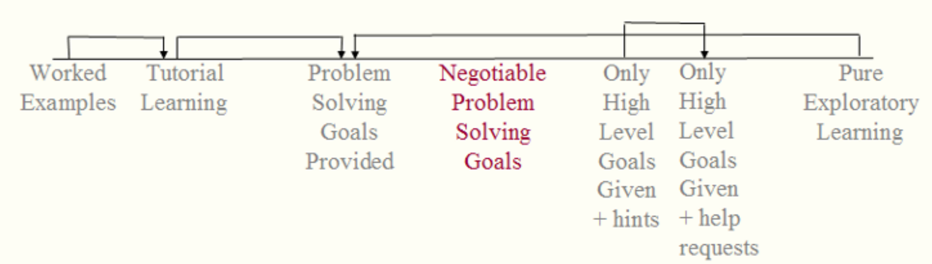

Worked Examples
Tutorial Learning
Problem Solving Goals Provided
Negotiable Problem Solving Goals
Only High Level Goals Given + hints
Only High Level Goals Given + help requests
Pure Exploratory Learning

Figure 1 The Exploratory Learning Continuum: Arrows connect approaches that have been experimentally compared in published works. The arrow points from the less effective approach towards the approach shown to be more effective.

Previous investigations of exploratory learning have compared student learning in conditions such as (1) passive worked example studying, (2) active but totally guided tutorial learning, (3) problem solving, and (4) unguided exploration. On the macro-level, what is manipulated is the amount of structure provided for students. In (1) and (2), for example, students make no choices whatsoever, although students in (2) are more active than students in (1). High level goals are set, and low level steps are provided. In (3), problem solving goals are made for the student, but the student chooses how to satisfy those goals through means-ends analysis. In (4), the student sets problem solving goals and chooses how to satisfy those goals. Thus, in (4) the student has the greatest autonomy, but the student is limited by their own conception of what is possible and valuable to explore. In (3) the student is prompted to explore areas in the space of possibilities that they may not have thought of by themselves. Furthermore, they reap the benefits of exploring alternative ways of achieving those goals. However, they do not get the practice setting goals for themselves that students in (4) get.

Many state-of-the-art tutoring systems fall into the problem solving category where problem solving goals are dictated. It is no cooincidence since published investigations along the Exploratory Learning Continuum have typcially shown this place on the continuum to be particularly effective. For example, Charnay & Reder (1986) compare Worked Examples, Tutorials, Problem Solving, and Pure Exploration. Worked examples mixed with problem solving was the best combination, consistent with other similar published results [16]. Along similar lines, Klahr & Nigam (2004) have shown in an empirical investigation of children learning the scientitic method that tutorial based learning mixed with problem solving is more efficient than pure exploratory learning. Other work has explored a part of the continuum in between problem solving and pure exploratory learning. In the light of a series of previous results showing the benefits of guided exploration over pure exploration [e.g., 9], the Smithtown work [e.g., 15] and the

Computer-Based Simulation Games work [10] involve guidance provided by high level goals such as learning about a model or survival in a simulation environment. Leutner (1993) demonstrates the importance of students with prior domain instruction actively requesting help rather than help being provided in an unsolicited manner during their interaction with a simulation environment. Note that in contrast to other published results that consistently point towards problem solving as the most promising point on the Exploratory Learning Continuum, these results point in the opposite direction, towards a less strongly guided approach, although they do not explicitly evaluate these two approaches in comparison with problem solving. In this paper we emipirically evaluate a new place on the continuum that we refer to as Negotiable Problem Solving Goals, which falls in between problem solving and the types of guided exploratory learning evaluated in the past [e.g., 9,15]. Our empirical investigation compares Negotiable Problem Solving Goals with two approaches that mix tutorial learning and problem solving. In all three conditions, students interact with a simulation environment.

Related to the distinction between "high level goals" and "low level goals" is the distinction between "learning oriented goals" and "performance oriented goals", which is the second conceptualization of exploratory learning that we investigate in this paper [3,11]. Some have argued that the distinction is identical and that under specified goals are inherently more learning oriented and correspondingly more conducive to learning. Others have argued that learning orientation is more of a characteristic of the learner than the task, and that even in connection with the same goals provided to the learner, learners with different orientations will approach the problem differently, and that difference in orientation may be responsible for the contributing to or detracting from the depth with which the learner absorbs the material [11].

2. Method

We are conducting our research in the domain of thermodynamics, using as a foundation the CyclePad articulate simulator [4]. CyclePad offers students a rich, exploratory learning environment in which they apply their theoretical thermodynamics knowledge by constructing thermodynamic cycles and performing a wide range of efficiency analyses without expense or danger.

Materials. The domain specific materials used in the study, which consisted of a take-home assignment, pre/post test, introductory reading material about rankine cycles, and focused readings with suggested illustrative analyses to perform using the CyclePad simulator for three forms of rankine cycles, were all developed by a Carnegie Mellon University mechanical engineering professor with the help of three of his graduate students and minimal input from our team. These domain specific materials were exactly the same across conditions, with the exception of the manipulation specific instructions described below. Thus, we strictly controlled for information presentation in all written materials. Additionally, we used a questionnaire to assess student attitudes after their participation.

Experimental procedure common to all conditions. The study consisted of two labs involving work with CyclePad that were assigned to the whole class. The first lab was a self-paced take-home assignment done during the first week of the study. The second lab was a 3-hour on-campus lab session completed during the second week of the study. Although the labs were mandatory assignments, participation in the study was optional. We strictly controlled for time between conditions. The 3-hour lab session was divided into 8 segments: (1) After completing the consent form, students were given 20 minutes to work through a 50 point pre-test consisting of short answer and multiple choice questions covering basic concepts related to rankine cycles, with a heavy emphasis on understanding dependencies between cycle parameters. (2) Students then spent 15 minutes reading an 11 page overview of basic concepts of rankine cycles. (3) Next they spent 25 minutes working through the first of three focused materials with readings, suggested problem solving goals, and analyses to help in meeting those goals. (4) Next they spent 20 minutes working through the second set of focused materials. (5) They then spent 20 minutes through the third set of focused materials. (6) They then spent 40 minutes in a Free Exploration phase creating the most efficient rankine cycle they could with no instructional support either from the tutor or any of the instructional materials they had been given previously. (7) They then spent 20 minutes taking a post-test that was identical to the pretest. (8) Finally, they filled out the questionnaire. The experimental manipulation took place during steps (3)-(5).

Experimental design. Our experimental manipulation consisted of 6 conditions resulting from a 3X2 full factorial design contrasting 3 goal level conditions and two goal orientation conditions. The three goal level conditions included (1) Negiatiable Problem Solving Goals (NPSG), which was human tutoring support + written materials that we refer to as a Script, (2) Problem Solving (PS), which consisted of help provided in the style of typical model tracing tutors + Script, and (3) Script only (S). In the Human tutoring condition (NPSG), students are given the opportunity to take the most initiative. In that condition they are free to select problem solving goals from the list provided, with some guidance from the human tutors, and to select from a provided list of CyclePad analyses to meet those goals. In the problem solving condition (PS), students follow the list of provided problem solving goals in order, but they decide how best to meet those goals from the suggested analyses. In the script condition (S), students follow the list of provided problem solving goals and achieve them by following the list of suggested analyses in the specified order. However, the instructions are specified at a high enough level that some means-ends analysis is still required to successfully follow them. Our goal orientation manipulation was a replication of [11], and was completely determined by manipulation specific instructions, which are described below.

It is important to note that the superiority of the human tutoring based negotiable problem solving goals condition is not a foregone conclusion in the light of recent results in the tutorial dialogue community, and thus presents a valid test of our hypothesis about negotiable problem solving goals. Consider the following series of empirical investigations. First, two evaluations of the AutoTutor system, in the domains of computer literacyand physics, showed an advantage over re-reading of the textbook of about 0.5 standard deviations [12,7]. The textbook re-reading condition itself was no better than a no-treatment control condition. However, in a different experiment the learning results obtained with WHY-AutoTutor were no worse than *a human tutoring condition* and yet not better than those in a control condition in which students read targeted "mini-lessons," short texts that covered the same content as that presented in the dialogue [6]. The mini-lesson condition is different from reading textbook text in that mini-lessons tend to be focused specifically on the knowledge and potential misconceptions involved in a specific exercise. It appears to be a high standard against which to compare. Note that the experimental procedure in our study involves extensive reading for students in all conditions. As a result, our experimental results can be seen as contributing to this line of investigating the trade-offs between human tutoring and a reading control. However, in order to place our experiment accurately in the context of previous results, it is important to consider the following differences. First, students in all conditions in our study were presented with exactly the same reading materials. Rather than replacing the reading materials as in [6], the role of the human tutors in our study was to help students navigate and understand the materials. Secondly, the reading materials were neither as brief nor targeted to the test as the "minilessons" nor were they as extensive as a text-book.

Outcome Measures: We looked at three outcome measures of instructional effectiveness. Two outcome measures were assessed by means of a Pre/Post test. 32 multiple choice and short answer questions were used to test analytical knowledge of Rankine cycles, including relationships between cycle parameters. An important aspect of this was a set of prediction questions where students were told to predict the impact of a specific change in one cycle parameter on several other cycle parameters. The other part of the test was a set of 9 open response questions assessing conceptual understanding of Rankine cycles. The third type of outcome measure we looked at was ability to apply knowledge to build and optimize a Rankine cycle using CyclePad during a Free Exploration phase.

Participants. We conducted our study over a two week period of time as part of a sophomore Thermodynamics course at Carnegie Mellon University beginning the week when Rankine cycles were introduced in the lecture portion of their class. Each student in the two NPSG conditions (NPSG-LO and NPSG-PO) who completed the study was tutored by one of three mechanical engineering graduate students during an individual tutoring session. The students in the other 4 conditions (PS-LO, PS-PO, S-LO, and S-PO) completed their 3-hour lab in a group lab session that was specific to their condition. Students were assigned to conditions in such a way as to maximize the evenness in distribution of grade so far between conditions and to respect student availability during 4 lab session times, as indicated on an on-line questionnaire. The average grade so far in the class for each condition was virtually identical. However, only 67 out of 120 students both attempted the take home lab and participated in the experiment. An additional 30 students completed the second lab but did not do the take-home assignment.

Manipulation specific instructions. Prior to the second lab, students were either told they were assigned to a specific group lab time or that they were to make an appointment for an individual lab time, but they were not told prior to the second lab what type of instructional treatment to expect or how their treatment differed from that of other students. In between segments (1) and (2) and also between segments (2) and (3) of the experimental procedure, students in the Learning Orientation (LO) condition were told that their goal was to learn as much thermodynamics as possible during the lab, and that at the end they would be asked to demonstrate the deep understanding that they acquired. In contrast, students in the Performance Orientation (PO) condition were told that their goal was to achieve the greatest cycle efficiency as possible and that in the end they would be asked to demonstrate their ability to achieve the greatest efficiency possible. Additionally, in between segments (2) and (3) students received instructions specific to their goal level manipulation.

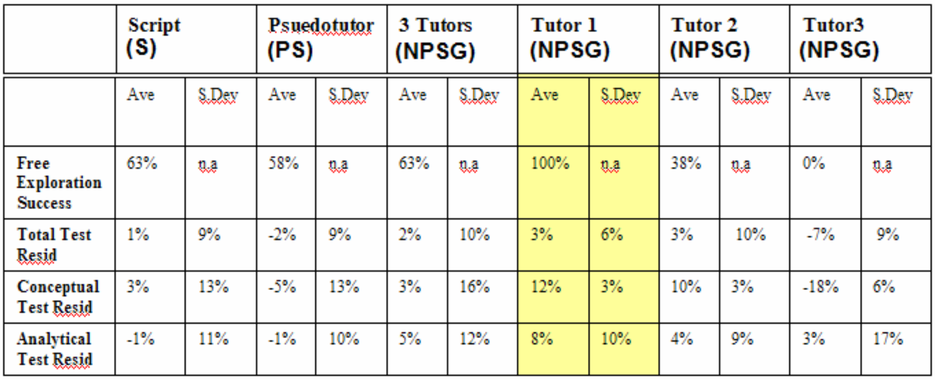
3. Results

First, we verified that the goal orientation manipulation had an effect on student goal orientation. We examined patterns of student responses on two goal orientation manipulation check questions on the questionnaire that were adapted from previous studies investigating student goal orientation [e.g., 3]. In both cases, students in the Learning Oriented (LO) condition were more likely to select the learning oriented response than students in the Performance Orientation condition (PO). We evaluated the reliability of the difference in proportion between conditions using a multinomial logistic regression. In the case of the first question, the difference was marginal t=1.58, p=.11. For the other question, the difference was significant, t=2.33, p<.05. Thus, we concluded that the goal orientation manipulation had an effect on the student population, and if differences in goal orientation do have an impact on student behaviour and learning, we should be able to detect these differences between conditions by examining our outcome measures.

Since not all students who participated in the 3 hour lab completed the take home assignment, we checked to see whether not having completed the assignment had an effect on how successful students were in learning during the lab. There was no significant difference in grade so far in the course between the students who participated in the lab and those who did not. Students who did not do the take-home assignment were evenly distributed across conditions. On average, it was the best students in the class who did the take-home assignment: Mean(no) = 70.26, s.d.= 11.7, Mean(yes)=75.5, s.d.= 9.2, t(95)=2.4, p<.05. However, controlling for pretest score, there was no reliable difference in post test score between students who did the take-home assignment and those who did not using a 2-tailed paired t-test, t(24)=1.12, p=.27. Thus, we considered students who did not do the take-home assignment in our analysis of learning gains on the Pre/Post test but not on our assessment of performance with CyclePad during the Free Exploration phase.

We found that there were serious problems with one of our three tutors, namely Tutor 3. He was extremely terse and impatient with students. His transcripts contained almost no conceptual discussion, and in his impatience, he rarely let students complete their work. Instead, he tended to take over and do the lab for them through the VNC connection to their simulation interface. Students who worked with him learned much less than expected based on their pretest scores, as clearly demonstrated in Table 1. Thus, we left the data from the students that he tutored out of the learning gains analysis described below.

Table 1 Overview of Outcome Measures from Goal Level Manipulation. Note that test results are reported in terms of residuals resulting from a regression between pretest and posttest score. In other words, this is the portion of the post test score that differs from what is expected purely based on pretest score. A positive value indicates how much higher the post-test score is over and above what is expected based on pre-test score, based on the pattern observed over the whole population. A negative value indicates how much lower the post-test score is than what is expected based on pre-test score.



	Script (S)		Psuedotutor (PS)		3 Tutors (NPSG)		Tutor 1 (NPSG)		Tutor 2 (NPSG)		Tutor3 (NPSG)	
	Ave	S.Dev	Ave	S.Dev	Ave	S.Dev	Ave	S.Dev	Ave	S.Dev	Ave	S.Dev
Free Exploration Success	63%	n.a	58%	n.a	63%	n.a	100%	n.a	38%	n.a	0%	n.a
Total Test Resid	1%	9%	-2%	9%	2%	10%	3%	6%	3%	10%	-7%	9%
Conceptual Test Resid	3%	13%	-5%	13%	3%	16%	12%	3%	10%	3%	-18%	6%
Analytical Test Resid	-1%	11%	-1%	10%	5%	12%	8%	10%	4%	9%	3%	17%

Overall there was a main effect for the Goal Level manipulation (F(2,83) = 3.81, p < .05, MSE = 20.9), but no main effect for Goal Orientation manipulation or the interaction between the two. Overall the order was PS < S < NPSG. Using a Bonferroni post-hoc analysis, we determined that the difference between NPSG and PS was significant (p <

.05), whereas the difference between NPSG and S was marginal (p=.11). The difference between the S and PS was only a statistical trend. Despite our dissapointment at having to drop the data from Tutor 3, we consider the stark difference in effectiveness between his tutoring and the other two tutors as an indication that it was the Goal Level manipulation and not just a "warm body" effect (i.e., that students just prefered working with a human tutor) that lead to the significant main effect for the Goal Level manipulation.

Because of larger differences in standard deviation within sections on the test than overal, the differences between conditions were less clear within individual test sections. On the conceptual part of the test, there was a significant main effect for Goal Level manipulation but not Goal Orientation manipulation, and no interaction effect. Again the order was PS < S < NPSG. Using a Bonferroni post hoc analysis, we determined that both S and NPSG were significantly better than PS (p < .05), whereas the difference between NPSG and S was only a trend (p=.16). On the objective part of the test there was no main effect either for Goal Level manipulation or Goal Orientation manipulation. However there was a marginal crossover interaction F(2,83) = 2.98, p=.06. The crossover interaction was between the P and PS conditions where PS was better in the Performance orinetation condition (PO), but S was better in the Learning Oriented Condition (LO).

We then evaluated student performance on the Free Exploration assessment. There we found no main effect for Goal Level manipulation or Goal Orientation manipulation overall, nor an interaction. However, we found a significant difference in effectiveness between tutors within the NPSG condition using a binomial logistic regression (p < .005). For Tutor 1, 100% of his students were able to successfully complete the Free Exploration portion of the assignment. For Tutor 2, only 36% of his students were able to complete it. For Tutor 3, whose data was thrown out of the learning gains analysis, 0% of his students were able to complete the free exploration portion of the lab. 58% of PS students and 63% of S students were able to complete it. Obviously, Tutor 1, as the best performing representative of the NPSG condition, was significantly more effective than the other tutors as well as the other Goal Level manipulations on this assessment.

Overall, we found significant Goal Level manipulation effects, with NPSG being the clearest win across the three outcome measures, especially Tutor 1, as displayed in Table 1. However, in contrast to findings in McNeil & Alibali (2000), we found very little evidence of any Goal Orientation effect.

4. Conclusions and Current Directions

The results of our empirical investigation offer strong support that a tutorial dialogue system based on the idea of Negotiable Problem Solving Goals for support in an exploratory learning environment is a promising new direction for the tutorial dialogue community. One common pattern that we have observed is that students start out with the idea that more sophisticated designs will be more efficient. Thus, students have a tendency to be drawn towards the more advanced portions of the design space before they are ready to fully understand how to use that sophistication to an efficiency advantage. When our tutors observe this behavior, they encourage students to keep it simple and direct them back to more basic design explorations until students demonstrate a solid understanding at that basic level. This high level structuring provides many of the advantages of previously explored problem solving conditions. Because of it, students are not hampered by their preconceptions that would have lead them to spend their time in explorations that would have been devoid of educational value. Yet, students in the NPSG condition take more initiative than in the S or PS conditions because they still have a hand in deciding how they will spend their exploratory time. We are currently conducting an in-depth corpus analysis to gain deeper insights into what lead to the differences in effectiveness between Tutors 1, 2, and 3 within the NPSG condition. We plan to use that analysis as the foundation for the CycleTalk tutorial dialogue system, which we are developing [14].

Acknowledgements

This project is supported by ONR Cognitive and Neural Sciences Division, Grant number N000140410107.

References

[1] Aleven V., Koedinger, K. R., & Popescu, O. (2003). A Tutorial Dialogue System to Support Self-Explanation: Evaluation and Open Questions. *Proceedings of the 11th International Conference on Artificial Intelligence in Education*, AI-ED 2003.

[2] Charney, D.H., & Reder, L.M. (1986). Designing tutorials for computer users: Effects of the form and spacing of practice on skill learning. *Human Computer Interaction*, 2, 297-317.

[3] Dweck, C. S., & Leggett, E. L. (1988). A social-cognitive approach to motivation and personality. *Psychological Review*, 95, 256-273.

[4] Forbus, K. D. (1999). CyclePad: An Articulate Virtual Laboratory for Engineering Thermodynamics. *Artificial Intelligence* 114(1-2): 297-347.

[5] Graesser, A. C., Bowers, C. A., Hacker, D.J., & Person, N. K. (1998). An anatomy of naturalistic tutoring. In K. Hogan & M. Pressley (Eds.), *Scaffolding of instruction*. Brookline Books.

[6] Graesser, A., VanLehn, K., the TRG, & the NLT. (2002). *Why2 Report: Evaluation of Why/Atlas, Why/AutoTutor, and Accomplished Human Tutors on Learning Gains for Qualitative Physics Problems and Explanations*, LRDC Tech Report, (2002) University of Pittsburgh.

[7] Graesser, A. C., Jackson, G. T., Mathews, E. C., Mitchell, H. H., Olney, A., Ventura, M., Chipman, P., Franceschetti, D., Hu, X., Louwerse, M. M., Person, N. K., and the Tutoring Research Group, (2003). Why/AutoTutor: A Test of Learning Gains from a Physics Tutor with Natural Language Dialog. *Proceedings of the Cognitive Science Society*.

[8] Klahr & Nigam (2004). The equivalence of learning paths in early science instruction: effects of direct instruction and discovery learning, *Psychological Science*, 2004.

[9]Leutner, D. (1993). Guided discovery learning with computer-based simulation games: effects of adaptive and non-adaptive instructional support. *Learning and Instruction*, 3, 113-132.

[10] Mayer, R. E. (2004). Should there be a three-strikes rule against pure discovery learning? The Case for Guided Methods of Instruction, *American Psychologist* 59(1), pp 14-19.

[11] McNeil, N. M., & Alibali, M. W. (2000). Learning mathematics from procedural instruction: Externally imposed goals influence what is learned. *Journal of Educational Psychology*, 92, 734-744.

[12] Person, N., Bautista, L., Graesser, A., Mathews, E., & The Tutoring Research Group (2001). In J. D. Moore, C. L. Redfield, & W. L. Johnson (Eds.), *Artificial Intelligence in Education: AI-ED in the Wired and Wireless Future, Proceedings of AI-ED 2001* (pp. 286-293). Amsterdam, IOS Press.

[13] Rosé, C. P., Jordan, P., Ringenberg, M., Siler, S., VanLehn, K., & Weinstein, A. (2001). Interactive Conceptual Tutoring in Atlas-Andes, In J. D. Moore, C. L. Redfield, & W. L. Johnson (Eds.), *Artificial Intelligence in Education: AI-ED in the Wired and Wireless Future, Proceedings of AI-ED 2001* (pp. 256-266). Amsterdam, IOS Press.

[14] Rosé, C. P., Torrey, C., Aleven, V., Robinson, A., Wu, C. & Forbus, K. (2004). CycleTalk: Towards a Dialogue Agent that Guides Design with an Articulate Simulator, *Proceedings of the Intelligent Tutoring Systems Conference.*

[15] Shute, Valerie J. and Glaser, Robert, (1990). "A Large Scale Evaluation Of An Intelligent Discovery World: Smithtown", *Interactive Learning Environments* 1, 51-77.

[16] Tuovinen, J. E., & Sweller, J. (1999). A comparison of cognitive load associated with discovery learning and worked examples. *Journal of Educational Psychology*, 91(2), 334-341.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Automatic and Semi-Automatic Skill Coding With a View Towards Supporting On-Line Assessment

Carolyn ROSÉ, Pinar DONMEZ, Gahgene GWEON, Andrea KNIGHT, Brian JUNKER, William COHEN, Kenneth KOEDINGER
Carnegie Mellon University,
5000 Forbes Avenue, Pittsburgh PA, 15213

Neil HEFFERNAN
Worcester Polytechnic Institute
100 Institute Road, Worcester MA, 01609-5357

Abstract. This paper explores the problem of automatic and semi-automatic coding of on-line test items with a skill coding that allows the assessment to occur at a level that is both indicative of overall test performance and useful for providing teachers with information about specific knowledge gaps that students are struggling with. In service of this goal, we evaluate a novel text classification approach for improving performance on skewed data sets that exploits the hierarchical nature of the coding scheme used. We also address methodological concerns related to semi-automatic coding.

1. Introduction

The goal of the TagHelper project [5] is to develop text classification technology to address concerns specific to classifying sentences using coding schemes developed in support of educational research an other behavioral research fileds. A wide range of behavioral researchers including social scientists, psychologists, learning scientists, and education researchers collect, code, and analyze large quantities of natural language corpus data as an important part of their research. Currently there are a wide range of corpus analysis tools used to support corpus analysis work either at a very low level (e.g., word frequency statistics, collocational analyses, etc.) or at a high level (e.g., exploratory sequential data analysis once a corpus has been coded with a categorical coding scheme), but no widely available tools to partly or fully automate the time consuming process of doing the categorical behavioral coding or content analysis. In this paper, we address both technical and methodological concerns in developing technology for streamlining the categorical type of protocol analysis.

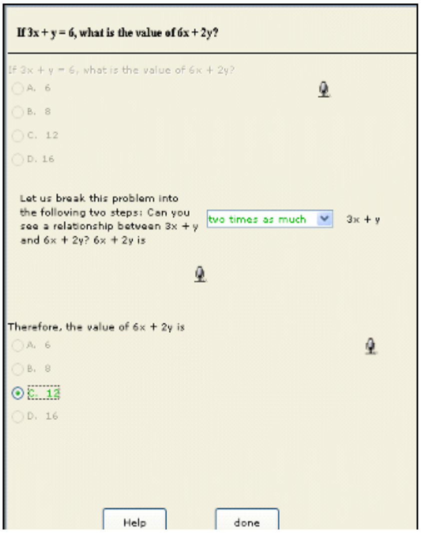
As an additional focus, in this paper we explore the potential of supporting on-line assessment with technology for automatic and semi-automatic skill coding of assessment items based on predictions from the text of the problem statements. On this level, the work reported in this paper is part of a larger effor towards addressing the “Assessment Dilema”, which is a fundamental dilemma teachers face in trying to use assessment to guide instruction. Specifically, assessment takes time away from instruction and teachers cannot be sure the time spent assessing will improve instruction enough to justify the cost of lost instructional time. We are addressing this dilemma by building and experimentally evaluating the effectiveness of a web-based "Assistment" system for middle school math in Massachusetts. On-line testing systems that grade students and provide reports reduce the demands on the teacher. However, they do not fundamentally address the assessment dilemma. In contrast to previous approaches, the Assistment system aims to 1) quickly predict student scores on standards-based tests, 2) provide feedback to teachers about how they can specifically adapt their instruction to address student knowledge gaps, and 3) provide an opportunity for students to get intelligent tutoring assistance as assessment data is being collected. Assistments provide more focused instruction than the feedback that is typically given by on-line multiple-choice systems. A skill coding of assessment items is meant to facilitate assessment of student knowledge on individual skills. The resulting model of student mastery can then be used for predicting total scores on standards based tests as well as mastery on individual standards. A detailed assessment of student knowledge is meant to keep teachers informed about the individual needs of their students in order to support them in their task of preparing their students for the tests.

In the remainder of the paper we discuss in greater depth how a skill coding of assessment items can be used to facilitate on-line assessment. We then discuss alternative coding schemes we have been exploring. Next we discuss recent success in fully automatic skill coding using the 39 Massachussetts state standards for math at the 8th grade level (MCAS39). We also present results from an empirical evaluation of a coding interface that demonstrates the impact of automatic predictions on coding speed, reliability, and validity for semi-automatic skill coding. We conclude with discussion of current directions.

2. Motivation for Skill Coding for Assessment

The purpose of coding math problems with required skills is to eventually allow us to compute predictions about performance on state exams based on a limited number of interactions with the Assistments system (e.g., approx. 20 minutes per week). This is still work in progress. One of our planned approachs is to track each student's progress on the multiple skills and other cognitive components needed to do well on state tests, through a fully multidimensional IRT model or Bayesian inference network (e.g., 13) based on Assistment data. From this, one can predict the student's performance on a set of test questions tapping a distribution of skills similar to that seen in past state assessments. However, state tests are largely still developed using unidimensional IRT as a scaling tool [e.g. 10,8], which tends to force most individual differences to be driven by total test score. While there have been some successes developing multidimensional diagnostic reports for national tests such as the PSAT/NMSQT [4], our preliminary work with MCAS historical data suggests that fine-grained individual differences are swamped by gross number-correct groupings of students on high-stakes state tests, making multidimensional prediction problematic.

We are developing a cognitively-based, state-independent representation for encoding mathematical competency. This representation will be used to code state learning objectives, state test items, whole Assistment items and individual Assistment scaffolds. This coding then serves multiple functions within the proposed infrastructure. First, it allows us to draw correspondences between state standards and those of other states as well as the NCTM standards from which they are derived. As a byproduct, it allows us to match individual Assistment items to the corresponding NCTM standards as well as individual state standards. The proposed representation is finer grained than typical state standards. Thus, we argue that it is more suited to the task of predicting item difficulty because it explicitly represents the factors that make an item either difficult or easy for students.



If 3x + y = 6, what is the value of 6x + 2y?

If 3x + y = 6, what is the value of 6x + 2y?

○ A. 6

○ B. 8

○ C. 12

○ D. 16

Let us break this problem into the following two steps: Can you see a relationship between 3x + y and 6x + 2y? 6x + 2y is [two times as much] 3x + y

Therefore, the value of 6x + 2y is

○ A. 6

○ B. 8

● C. 12

○ D. 16

Figure 1 – Sample Assistment Item

	Overlap	Unique Non-Overlap
NCTM	"Use symbolic algebra to represent situations and to solve problems, especially those that involve linear relationships."	
MCAS P.7	"Set up and solve linear equations and inequalities with one or two variables, using algebraic methods, models, and/or graphs."	- specifies number of variables - models and graphs in addition to algebraic expressions
PSSA 2.8.8.E **(algebra strand)**	"Select and use a strategy to solve an equation or inequality, explain the solution and check the solution for accuracy."	- explaining the solution - checking the answer

Figure 2 Non-overlap of individual state learning objectives

While the state standards for mathematics nationwide are all based on the NCTM standards for mathematics, the example problem in Figure 1 illustrates why a state-independent component representation of mathematical knowledge is required for generalizing across state standards. Figure 2 displays the non-overlap between the relevant NCTM standard and the relevant learning objectives for Massachusetts (MCAS) and that of Pennsylvania (PSSA) for that problem. Because of the lack of direct correspondence between individual standards for different states as well as between NCTM standards and state specific standards, a more basic and fine grained representation is needed to demonstrate the precise connection between these different but very strongly related systems of standards.

A key characteristic of our cognitively-based knowledge representation is that it is composed of a vector of learning factors that distinguish problems from one another and predict item difficulty based on scientific findings from prior research and available state test results. An example of a learning factor is that students are known to have more trouble with scatter plots than line graphs partly because they are less common [1]. However, even important distinctions do not apply to all types of problems. For example, the graph type factor only applies to problem types that include graphs. In order to limit the number of judgments required to assign values to the representation vector for a specific item by human coders, we have designed a two-level representation in which first order learning factors identify the problem type (e.g., graph interpretation problems, simple algebraic simplification problems, or linear equality problems), and second-order learning factors make more fine grained distinctions (e.g., which type of graph, complexity of symbolic representation, or number of variables involved). Once the first-order factors have been specified, only a subset of second-order factors are relevant, and the others can be assigned a default value automatically.

3. Explorations of Fully Automatic Skill Coding

As we have been developing our cognitively based coding scheme, we have been exploring automatic coding with existing skill codings such as the MCAS39 as a proof-of-concept. The data we have consists of multi-class labels. There are 154 instances and 39 codes where each instance can be assigned a subset of these 39 codes. These codes are formed by 5 general categories; G, N, M, P, and D. Each of these categories has sub-level categories; for instance D-category is regarded as D.1, D.2, D.3, and D.4.

Applying a categorical coding scheme can be thought of as a text classification problem where a computer decides which code to assign to a text based on a model that it has built from examining "training examples" that were coded by hand and provided to it. A number of such statistical classification and machine learning techniques have been applied to text categorization, including regression models [12], nearest neighbor classifiers [12], decision trees [9], Bayesian classifiers [6], Support Vector Machines [7], or rule learning algorithms [2]. While these approaches are different in many technical respects that are beyond the scope of this paper to describe, they are all applied the same way. A wide range of such machine learning algorithms are available in the Minorthird text-learning toolkit [3], which we use as a resource for the work reported here.

One challenge in applying text classification technology to word problems is that the text of word problems contain many superficial features that make texts appear similar when they are very different at a deep level, or conversely, different when they are very similar at a deep level. These features include numbers, fractions, monetary values, percentages, dates, and so on. Thus, we replaced all the occurrences of features mentioned above with some pre-defined meta-labels, such as number, fraction, date, etc. A wide range of simple replacements can be made easily using search-and-replace facilities provided by the MinorThird toolkit. Other more complicated features must be tagged by hand and then trained using text classification technology.

As a baseline for our evaluation we explored training a binary classifier for each code using 4 standard text classification algorithms; namely SVM, DecisionTree, NaiveBayes, and VotedPerceptronLearner. In particular, SVM and VotedPerceptron classifiers are known to perform well on skewed data sets such as ours. We compared their performance using a

10-fold cross-validation methodology. SVM was the best performing approach. Nevertheless, although the performance was high in terms of percent correct, agreement with the gold stadard measured in terms of Kappa was very low, frequently 0, and in some cases negative.

The novel text classification approach we explore in this paper, which is our primary technological contribution, exploits the hierarchical nature of the MCAS coding scheme. The basic idea involves dividing the whole corpus into clusters according to the general categories, and then training and testing a binary classifier within each cluster separately. The hypothesis behind this approach is that if we can obtain relatively homogeneous clusters by exploiting each general category, then it will be simpler to train classifiers to operate within clusters because there will be fewer distinctions to make. Furthermore, since the texts within a cluster will be similar to each other, the trained classifiers can hone in on the fine distinctions that separate the lowest level classes.

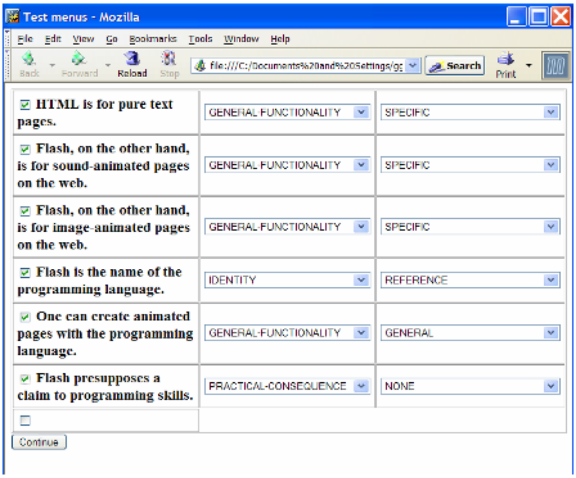
We used a 10-fold cross-validation methodology to train classifiers for splitting the data into clusters. For example, on each itteration, we train a classifier for each of the 5 general categories over 9/10 of the data. We then use the trained classifier to split the 10th segment into 5 separate clusters, one for each general category. We do this 10 times and then combine all of the separate clusters that belong to the same general category.Separation into clusters using the trained classifiers was not perfect. Nevertheless, the similarity between texts within clusters was still higher than over the whole corpus, and fewer separate low level classes were in each cluster than were in the whole set. We then used 10-fold cross-validation within clusters to obtain an accuracy for binary classifiers within clusters. We combined the results from individual clusters in order to obtain an agreement score for each of the MCAS39 labels across clusters using cluster specific classifiers.

On average the new classifiers performed significantly better than the baseline classifiers both in terms of percent agreement and Kappa ($p < .05$). Out of 29 classes that we had at least 2 instances of in our data, we were able to train classifiers to detect 13 of them at the .7 Kappa level or better. An additional 5 were between the .65 and .7 Kappa level, just missing an acceptable performance. An additional 5 showed significant improvement but did not reach the .7 level. For 4 out of the 29 classes, we were not able to achieve a substantial improvement over the baseline. In order to achieve an acceptable level of agreement while saving time over coding by hand, it is possible to allow the classifiers that have an acceptable performance be applied to the data and simply check the data over for places where additional codes from the remaining classifiers must be added. The first level classification of the data into rough clusters effectively narrows down the number of categories that must be considered for any single problem. Thus, we have determined that on average, using the information provided by the automatic predictions, a human coder would only need to consider 8 potential codes on average rather than 39 in order to achieve a complete coding of the data with human level agreement.

4. Issues Related to Semi-Automatic Skill Coding

While these explorations of automatic coding technology are promising, they leave open the question of what is the best course of action for dimensions of coding schemes where an acceptable level of agreement with a reliable gold standard cannot be achieved with a fully automatic approach. This is typically the case where there is a shortage of hand coded examples to use for training, or there are many categories that are very subtly differentiated, or there are many infrequently occurring categories. For example, the amount of hand coded data we had access to for the MCAS coding experiment described above was relatively small (only 150 instances). And several categories only occurred one or two times in the whole set. The question is whether it is better in cases where automatic coding cannot be done with an acceptable level of reliability to make automatic predictions, which will then be checked and corrected, or simply to code a portion of the data with no support of automatic predictions. To this end, we conducted a small formal study to measure the impact of automatic predictions on speed, validity, and reliability of human judgment when applying a categorical coding scheme.

Materials. For this study we use a coding scheme developed in connection with a net based communication project focusing on usage of technical terms in expert-layperson communication described in [11]. Materials for the experiment include (1) a 6 page coding manual that describes the definitions of a coding scheme with 14 separate codes and gives several examples of each; (2) a training exercise consisting of 28 example sentences; and finally, (3) 76 sentences for the experimental manipulation. Two expert analysts worked together to develop a "Gold Standard" of coding for the explanations used in the training exercises as well as the examples for the experimental manipulation that indicates the assigned correct code for each sentence.


Test menus - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back
Forward
Reload
Stop
file:///C:/Documents%20and%20Settings/g:
Search
Print
Continue

☑ HTML is for pure text pages.	GENERAL FUNCTIONALITY	SPECIFIC
☑ Flash, on the other hand, is for sound-animated pages on the web.	GENERAL FUNCTIONALITY	SPECIFIC
☑ Flash, on the other hand, is for image-animated pages on the web.	GENERAL-FUNCTIONALITY	SPECIFIC
☑ Flash is the name of the programming language.	IDENTITY	REFERENCE
☑ One can create animated pages with the programming language.	GENERAL-FUNCTIONALITY	GENERAL
☑ Flash presupposes a claim to programming skills.	PRACTICAL-CONSEQUENCE	NONE
☐		

Figure 3 Prototype TagHelper interface used in study

Coding interface. Participants coded the example sentences for the experimental manipulation using a menu-based coding interface displayed in Figure 3. For the standard coding interface used in the control condition, the example sentences were arranged in a vertical list on a web page. Next to each sentence was a menu containing the complete list of 14 codes, from which the analyst could select the desired code. No code was selected as a default. In contrast, a minimally adaptive version was used in the experimental condition. The only difference between the adaptive version and the standard version was that in the adaptive version a predicted code was selected by default for each sentence. That predicted code appeared as the initial element of the menu list and was always visible to the analyst. The other elements of the list in each menu were identical to that used in the standard version, so correcting incorrect predictions was simple.

Participants. The participants in our study were Carnegie Mellon University and University of Pittsburgh students and staff. 20 participants were randomly assigned to two conditions. In the control condition, participants worked with the standard coding interface described above. In the experimental condition, participants worked with the minimally adaptive coding interface described above that displays predicted codes for each sentence in the corpus set up in such a way that 50% of the sentences were randomly selected to agree with the Gold Standard codes, and the other 50% were randomly assigned. We randomly selected which sentences to make incorrect predictions about so that the distribution of correct versus incorrect predictions would not be biased by the difficulty of the judgment based on the nature of the sentence.

Experimental procedure. Participants first spent 20 minutes reading the coding manual. They then spent 20 minutes working through the training exercise using the coding manual. As they worked through the 28 example sentences, they were instructed to think aloud about their decision making process. They received coaching from an experimenter to help them understand the intent behind the codes. After working though the training exercise, participants were given a Gold Standard set of codes for the training sentence to compare with their own. Altogether training took 45 minutes. After the training phase, participants were given a five minute break. They then spent up to 90 minutes working through 76 sentences, coding each sentence.

First we evaluated the reliability of coding between conditions. Average pairwise Kappa measures were significantly higher in the experimental condition ($p < .05$). Mean pairwise Kappa in the control condition was .39, whereas it was .48 in the experimental condition. As a measure of the best we could do with novice analysts and 50% correct predicted codes, we also analyzed the pairwise Kappa measures of the 3 participants in each condition who's judgments were the most similar to each other. With this carefully chosen subset of each population, we achieved an average pairwise Kappa of .54 in the control condition and .71 in the experimental condition. This difference was significant ($p < .01$). The average agreement between these analysts' codes from the experimental condition and the Gold Standard was also high, an average Kappa of .70. Thus, the analysts who agreed most with each other also produced valid codes in the sense that they agreed with the Gold Standard. Next we evaluated more stringently the validity of coding. We found that analysts in the experimental condition were significantly more likely to agree with the prediction when it was correct (74% of the time) than when it was incorrect (16% of the time). This difference was significant using a binary logistic regression with 760 data points, one for each sentence coded in the experimental condition ($p<.001$). Average percent agreement with the gold standard across the entire population was significantly higher ($p < .05$), and average Kappa agreement was marginally higher in the experimental condition than in the control condition ($p=.1$). Average agreement in the unsupported condition was a Kappa measure of .48. In the experimental condition, average agreement with the gold standard was a Kappa measure of .56. Thus, we conclude that analysts were not harmfully biased by incorrect codes. Coding time did not differ significantly between conditions, thus providing some confirmation of the estimate that 50% correct predictions is a reasonable break even point for coding speed. Average coding time in the control condition was 67 minutes and 36 seconds. In the experimental condition average coding time was 66 minutes and 10 seconds. On average, time saved by checking rather than selecting a code was roughly equivalent to time lost by correcting a prediction after checking and disagreeing with a prediction.

5. Current Directions

In this paper we have discussed the problem of automatic and semi-automatic coding of on-line test items both from the language technology and human-computer interaction angles. The specific application area we discussed was a skill coding of math assessment items, the purpose of which is to allow the assessment to occur at a level that is both indicative of overall test performance on state exams and useful for providing teachers with information about specific knowledge gaps that students are struggling with. We presented results from an evaluation that demonstrates that skill coding of math assessment items can be partially automated and a separate formal study that argues that even in cases where the predictions cannot be made with an adequate level of reliability, there are advantages to starting with automatic predictions and making corrections, in terms of reliability, validity, and speed of coding. One focus of our continued research is developing new text classification techniques that work well with heavily skewed data sets, such as our MCAS coded set of math problems.

6. Acknowledgements

This work was supported by the National Science Foundation grant number SBE0354420 and a grant from the US Department of Education, Institute for Education Sciences, Effective Mathematics Education Research grant number R305K030140.

References

[1] Baker R.S., Corbett A.T., Koedinger K.R., Schneider, M.P. (2003). A Formative Evaluation of a Tutor for Scatterplot Generation: Evidence on Difficulty Factors. *Proceedings of the Conference on Artificial Intelligence in Education*, 107-115.

[2] Cohen, W. and Singer, Y. (1996). Context-sentsitive learning methods for text categorization, In *SIGIR'96: Proc. 19th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pp. 307-315.

[3] Cohen, W. (2004). *Minorthird: Methods for Identifying Names and Ontological Relations in Text using Heuristics for Inducing Regularities from Data*, http://minorthird.sourceforge.net.

[4] DiBello, L. and Crone, C. (2001, July). Enhanced Score Reporting on A National Standardized Test. *Paper presented at the International meeting of the Psychometric Society*, Osaka, Japan.

[5] Donmez, P., Rose, C. P., Stegmann, K., Weinberger, A., and Fischer, F. (to appear). Supporting CSCL with Automatic Corpus Analysis Technology, to appear in *the Proceedings of Computer Supported Collaborative Learning.*

[6] Dumais, S., Platt, J., Heckerman, D. and Sahami, M. (1998). *Inductive Learning Algorithms and Representations for Text Categorization*, Technical Report, Microsoft Research.

[7] Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features, In *Proc. 10th European Conference on Machine Learning (ECML)*, Springer Verlag, 1998.

[8] Massachusetts Department of Education (2003). *2002 MCAS Technical Report*. Malden, MA: Author. Obtained August 2004 from http://www.doe.mass.edu/mcas/2003/news/02techrpt.pdf

[9] Lewis, D. and Ringuette, R. (1994). A Comparison of teo learning algorithms for text classification, In *Third Annual Symposium on Document Analysis and Information Retrieval*, pp. 81-93.

[10] Mead, R., Smith, R. M. and Swandlund, A. (2003). *Technical analysis: Pennsylvania System of School Assessment, Mathematics and Reading.* Harrisburg, PA: Pennsylvania Department of Education. Obtained August 2004 from http://www.pde.state.pa.us/a_and_t/lib/a_and_t/TechManualCover.pdf.

[11] Wittwer, J., Nückles, M., Renkl, A. Can experts benefit from information about a layperson's knowledge for giving adaptive explanations?. In K. Forbus, D. Gentner, T. Regier (Eds.), *Proc. Twenty-Sixth Annual Conference of the Cognitive Science Society*, 2004. 1464-1469.

[12] Yang, Y. and Pedersen, J. (1997). Feature selection in statistical learning of text categorization, In *the 14th Int. Conf. on Machine Learning*, pp 412-420.

[13] Yan, D., Almond, R. and Mislevy, R. J. (2004). *A comparison of two models for cognitive diagnosis. Educational Testing Service research report #RR-04-02.* Obtained August 2004 from http://www.ets.org/research/dload/RIBRR-04-02.pdf

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

The Use of Qualitative Reasoning Models of Interactions between Populations to Support Causal Reasoning of Deaf Students

[1]Paulo SALLES, [2]Heloisa LIMA–SALLES, [3]Bert BREDEWEG
{[1]Institute of Biological Sciences, [2]Department of Linguistics},
University of Brasília, Campus Darcy Ribeiro, 70.919-900 Brasília, DF, Brazil;
[3]Human Computer Studies Laboratory, Faculty of Science, University of Amsterdam,
Kruislaan,419, Matrix I, 1098 VA Amsterdam, The Netherlands

{psalles@unb.br; hsalles@unb.br; bredeweg@science.uva.nl}

Abstract. Making inferences is crucial for understanding the world. The school may develop such skills but there are few formal opportunities for that. This paper describes an experiment designed to investigate the use of qualitative reasoning models to support deaf students in making inferences about the behaviour of populations in interactions such as commensalism, amensalism, and predation. The experiment was done in two sessions. In both, the teacher presented the concepts, which were translated to the signed language, and at the end the students answer to a test, consisting of objective questions and a written essay. In the second session qualitative models about the same types of interactions were used to show the structure of the two populations system and the dynamics of the system over time. Statistical analysis showed that the use of qualitative models had a significant positive effect on the performance of the students. They gave more correct answers to objective questions and produced less trivial conclusions in their essays. We are confident that qualitative models have an important role to play in their scientific education and in the acquisition of Portuguese as a second language.

1 Introduction

Inferences are fundamental for the comprehension of the world. It is a natural ability, but education may improve this capacity, by rendering it explicit. For those with special needs, like deaf students, there are some additional requirements. Brazilian deaf students are being integrated in the classroom with non-deaf students and have to acquire Portuguese as their second language, being the Brazilian Sign Language (LIBRAS) legally recognized as their first language. Qualitative Reasoning [11] may be useful in this respect, providing visual oriented presentation of the models and explicit representations of causality, used to explain structure and behaviour of physical systems. An exploratory study about the use of qualitative models in science education to support second language acquisition by deaf students is presented in [7]. The work described here further explore the potential of qualitative models to support their ability of making inferences in the context of second language acquisition mediated by science education. The goal of the present study is to evaluate the impact of using qualitative models in making inferences about interacting populations [5], as addressed in biology classes, taking into consideration the linguistic performance of the deaf students using written Portuguese in two tests, which include answering objective questions and writing essays. We are also looking for evidences that the use of qualitative models may improve their ability to express causal reasoning in written

Portuguese. We discuss the linguistic performance of the students in terms of the notion of *relevance*, as formulated in [9]. According to these authors, "relevant information is information that modifies and improves an overall representation of the world". The students' linguistic performance is evaluated by assessing the number of conclusions they were able to derive that imply modification and improvement of the overall representation of the world. The paper is organized as follows: in the next section, we introduce basic notions of interactions between populations and explain how these issues were used to assess deaf students' abilities for making inferences. In section 3 we discuss the methodology used in the experiment. The results are presented in section 4 and we close with a discussion of the results and final considerations.

2 Interactions between populations: concepts, models and simulations

Interactions between populations of different species are an important subject in ecology and resource management, both for theoretical studies, for example, about the structure of communities, and for practical applications, such as the development of diseases and agricultural production related technologies. Species interactions can be classified according to combinations of the symbols {–,0,+}: the symbol '–' means that one population is negatively affected by the other; '0' means that one population is not affected by the other; and '+' means that one population is positively affected by the other population. Positive and negative effects may be understood as influences on population growth. Accordingly, the symbols {–,0,+} indicate that the population is respectively decreasing, stable or increasing due to the interaction with the other population [5]. Based on these ideas, Salles *et al.* [6] present a set of qualitative models about six different types of interactions between populations. Models about three types of interactions described in [6] and shown in Table 1 are used here to support causal reasoning by deaf students:

Table 1. Interactions between populations and the interpretations of their effects.

Interaction type	Representation	Interpretation of influences
Commensalism	(A,B) = (0,+)	If A changes, then B changes in the same direction; if B changes, A does not change.
Amensalism	(A,B) = (0,–)	If A changes, then B changes in the opposite direction; if B changes, A does not change.
Predation	(A,B) = (+,–)	If A changes, then B changes in the opposite direction; if B changes, then A changes in the same direction.

2.1 The ontology and the tools used in the experiment

We adopted the ontology provided by the Qualitative Process Theory [4]. Accordingly, changes in populations are explained as consequences of the effects of other populations on their basic processes of natality and mortality. Following this ontology, processes are modelled as direct *influences* (*I+ and I–*) of their rates on state variables, and the effects of processes propagate to other quantities via *qualitative proportionalities* (*P+ and P–*). Simulations were run in the qualitative simulator GARP [3] and inspected by using the GUI VisiGarp [2]. Figure 1 presents relevant information typically shown and discussed with learners during the experiment.

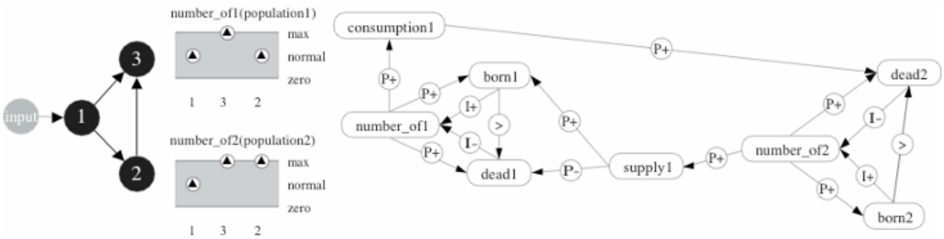


Figure 1: Simulation results for predation visualised by VisiGarp: state-graph (LHS), value-history (middle), and causal-model (RHS), being population 1 the predator and population 2 the prey.

2.2 The hypotheses tested in this study

Based on consultation with teachers and on the experience described in [7], we prepared the following research questions for this experiment: (RQ1) – Do qualitative models enhance understanding of causal relations in interactions between populations? (RQ2) – In interactions such as commensalism and amensalism, is it easier for the students to predict the effects of changes in the population caused by the other in utterances such as ['if A is increasing, then B is increasing'] than to recognize that changes in the latter (population B) do not influence the former (A), in utterances such as ['if B is increasing, then A does not change']? (RQ3) – In predation, is it easier for the students to predict how changes in the predator influence the prey population than to predict how changes in the prey population affect the predator population? (RQ4) – Is there any difference for the students to recognize the effects of positive and negative influences in interactions between populations? (RQ5) – Considering a food chain such as [A → B → C → D], is it easier for the students to predict changes propagated to the next level above or below (e.g. 'if C is increasing, then D is decreasing') than to predict changes propagated to organisms placed two or more levels above or below (e.g. 'if B is increasing, then D is increasing')? (RQ6) – Is there any difference for the students to answer questions about the interactions if the populations are identified in general terms (such as X and Y) instead of using their names? (RQ7) – Is it possible to find any difference in the occurrences of trivial and non-trivial conclusions in the written essays after the use of qualitative models?

3 Methodology

This study was developed in a secondary state school[1], with deaf students from the 2nd year. The experiment was run with the support of interpreters of LIBRAS-Portuguese who remained in the classroom during the tutorials. The experiment was set in two parts: (a) a session in 16/11/04, consisting of an oral presentation by a teacher, with an interpreter, followed by Test I; (b) a session in 25/11/04, consisting of an oral presentation, supported by qualitative models, with an interpreter, followed by Test II. During the experiment the teacher presented the effects of the interactions in terms of *if – then* utterances, and the students did not play with the models. Six deaf students participate in the first session and nine students in the second session[2]. Among them, six students participate in both sessions.

[1] This study was made in the same school where the experiment described in [7] was run.

[2] Three deaf students were involved in the previous study [7]. Two of them participated in both sessions and one student participated only in the second session in the present study. Due to the small number of

They are fluent in LIBRAS and have some mastering of (written) Portuguese as a second language, given their exposure to this language since their early (formal) education. As shown in the tests, the subjects display different levels of Portuguese, which will be abstracted away, as the present study is not concerned with comparing and (or) establishing their level of proficiency.

In the first session, a tutorial about interactions between populations was given to the students as they normally have in their school classes. It was explained that these interactions can be classified as beneficial (positive) and harmful (negative), depending on their effects on natality and (or) mortality. Next, the students were presented to examples of commensalism, amensalism and predation. Finally, concepts related to predation were explored in food chains involving well known animals and plants. Test I consisted of seven questions, designed to evaluate their ability in the following tasks: (a) to point out basic definitions of species, population and community; (b) to define benefit and harm; (c) to identify, in diagrams, the type of interaction by writing the name of the interaction or the sign of the influence in blank spaces; (d) to identify the effects of the interaction in each population; (e) to identify the consequences of changes in a population in a food chain with 3 organisms; (f) to identify the consequences of population changes in a food chain with 6 organisms; (g) to write an essay about the consequences of changes in a food web consisting of two food chains (6 and 5 organisms). Questions (a) to (f) included 30 items for the students to answer. All the questions but (a) asked for predictions about the consequences of a particular change in the system by using inferences as, for example, *IF population X is increasing, THEN population Y is decreasing*. The students should write *correct* / *incorrect*, and *increases* / *does not change* / *decreases* in blank spaces. In the written essay (g) the students were asked to explore formulations such as *IF X happens, THEN Y happens*, and *GIVEN THAT X happened, THEN Y will happen*.

In the second session, initially the students were exposed to a simple qualitative model for introducing vocabulary and modelling primitives [7]. Next, models about interactions between populations (commensalism, amensalism and predation) were presented to the students. In each case, an example involving well known organisms was given. A slide with a VisiGarp screenshot of the causal model was used to explain how benefit and harm were implemented. Finally, a simulation was run and a behaviour path (consisting of two or three states) was selected. Only values of the number of individuals in both populations were shown in the value history diagram. Changes in magnitudes and derivatives were pointed out as consequences of the interaction. In this session, no comments were made about food chains or food webs. Test II consisted of nine questions designed to evaluate the students ability (a) to understand the basic modelling primitives; (b) to understand representations of magnitudes and derivatives in the value history diagram; (c) to associate benefit and harm with their effects on natality and mortality; (d) to identify the effects of predation by writing *increases* / *decreases* in blank spaces; (e) to identify the effects of commensalism; (f) to identify the effects of amensalism; (g) to solve a problem involving the combination of predation and commensalism; (h) to predict the consequences of changes in a particular population on a food chain with 4 organisms; (i) to write an essay about the consequences of changes in a food web with 15 organisms consisting of three food chains with 4, 5 and 6 organisms. Questions (a) to (h) included 34 items for the students to answer, filling blank

deaf in the population and to difficulties in finding homogeneous groups of deaf students, this sample may be considered acceptable, if compared with other studies of the same kind.

spaces in a similar way as done in Test I. In the written essay (i) the students were asked to explore the same formulations used in Test I and a third one, *Y happens BECAUSE X had happened*. This experiment was not designed to assess learning based on pre and post-tests. Although exploring the same concepts, Test II was far more complex than Test I in many aspects, as for example, relating natality and mortality to benefit and harm, using terms such as X,Y sometimes replace the name of organisms, including a problem involving predation combined with commensalism, and exploring a more complex food web in the essay. Evaluation of the written essays consisted of identifying the manipulation of the concepts, in terms of the types of conclusions drawn by the students. Following [9], the conclusions were classified as trivial and non-trivial (see below). In order to test the significance of the results under the set of hypotheses presented in section 2, three nonparametric statistical tests were used: Mann-Whitney, Chi-square (χ^2) [8] and the test of significance for proportions [10]. Due to the similar results only the Chi-square results are presented here. The level of significance was defined in $\alpha = 0{,}05$.

4 Results and discussion

The use of qualitative models had a positive effect on the students' capacity of answering objective questions about interactions between populations (RQ1). The global analysis showed that the students gave significantly more correct answers in Test II than in Test I ($\chi^2 = 4{,}277$; 1 degree of freedom (df); P = 0,039). We believe that the use of qualitative models and simulations made the domain concepts clearer for them, probably because there was a visual representation of the structure of the system involving the two interacting populations, and of the behaviour of the system, shown as a sequence of states, with diagrammatic representations of the values of relevant quantities over time. Information represented this way is easily captured by deaf students, a result already found in [7]. However, the improvement was not homogeneously distributed among the students. Two out of six students that participate in both tests had significantly better performance and one had significantly worse performance in Test II. The other three students did not show significant improvement in their performance. Investigating whether or not the students would find more difficult to recognize that one population does not affect the other in commensalism and amensalism (RQ2), we found no significant differences in the results of Test I and Test II with respect to commensalism. However, the students gave significantly more correct answers in amensalism to questions involving utterances such as ['A causes change on B'] than to utterances such as ['B does not cause changes on A'] ($\chi^2 = 4{,}208$; 1 df; P = 0,040). In predation, the students gave significantly more correct answers to questions involving utterances of the type ['If the predator is increasing then the prey is decreasing'] than to questions involving utterances such as ['If the prey is decreasing then the predator is decreasing'] (RQ3). These results were observed both within Test I ($\chi2 = 8{,}853$; 1 df; P = 0,003) and within Test II ($\chi^2 = 11{,}815$; 1 df; P = 0,001). However, we found no significant differences between Test I and Test II when comparing correct answers to questions about both types of situations. Interesting to note that the results reported above about commensalism, amensalism and predation (RQ2 and RQ3) are not related to the students' abilities of recognizing benefit (positive influences) and harm (negative influences) in the three types of interactions (RQ4). The statistical analysis proposed in RQ2 showed no significant differences. A possible explanation for this difference may be the fact that the examples of commensalism explored in the two sessions are found in any textbook and are typically presented by teachers, while amensalism is not a well known interaction, and the students were not familiar with the examples used to illustrate such relation. In predation, in which causality is bidirectional, the starting point of the changes may produce very different results. For example, if the predator increases first, then the prey decreases, and if the prey decreases first, then the predator also decreases. Our study showed clearly that the students find more difficulties to identify changes in predator caused by changes in the prey. Maybe it has to do with their knowledge of the world. After all, young children notice that predators kill and eat their prey. Noticing that availability of food may cause effects in predator populations is more subtle. However, this is certainly an interesting point for further explore the potential of qualitative models. Also, we found no statistical support to the hypothesis that it is more difficult to predict propagation of changes to organisms placed two or more levels above or below than changes in organisms at the next level in a food chain (RQ5). It contradicts the results obtained in [7], in which the students found more difficulties to find the consequences of changes in the third position (Z) of the causal chain in utterances like ['If X is increasing, then Y is increasing and Z is decreasing']. Once again, the better performance here may be related to their familiarity with predation and food chains. We found no significant differences within Test I and within Test II with respect to the way the organisms involved in the interactions were identified, either by their names of by general terms such as X and Y (RQ6). However, in Test II the students gave significantly more correct answers to questions in which the organisms were identified by general terms (for example, X,Y) than in similar questions of Test I (χ^2 = 10,087; 1 df; P = 0,001). Although not conclusive, these results suggest that the students' capacity of dealing with abstract representations increased after the use of qualitative models, an issue to be explored in further work.

The linguistic performance of the students in the essays is discussed in terms of the notion of relevance, as formulated in [9]. As mentioned above, information that modifies and improves an overall representation of the world is considered to be relevant information. A representation of the world may in turn be regarded as a stock of factual assumptions and each newly acquired factual assumption is combined with the stock of existing assumptions to undergo inference processes whose aim is to modify and improve the individual's overall representation of the world. Factual assumptions are treated by the mind as true descriptions of the world. They are acquired from four sources: perception, linguistic decoding, assumptions and deductions. An assumption is a structured set of concepts to whose presence and structural arrangements deductive rules are sensitive. Concepts appear as an address in memory and may appear as a constituent of a logical form: "when the address of a certain concept appears in a logical form being processed, access is given to the various types of information stored in memory at that address" (cf. [9], p.86). The cognitive system monitors for redundancies and contradictions in its derivations, and the device continues to operate until no new theses can be derived. Improvements in the representation of the world are then traced via the workings of the human deductive device, which takes into account the semantic properties that are reflected in the form of assumptions. For the authors, the human deductive device has access only to elimination rules and yields only non-trivial conclusions. While introduction rules (for example, 'The Prime Minister has resigned') produce trivial conclusions (e.g. 'The Prime Minister has resigned and its warm today') in the sense that "they leave the content of their input assumptions unchanged (except for the addition of arbitrary material)", elimination rules (e.g. (i) input: P; (ii) If P then Q; (iii) output: Q) are genuinely interpretive, in the sense that "the output assumptions explicate or analyse the content of the input assumptions" (cf. [9], p. 97). A central function of the deductive device is to derive the contextual implications of any newly presented information in a context of old information [9]. Non-trivial conclusions are then directly derived, although the validity of arguments may be checked by procedures other than direct derivation. The deductive device is then expected to be complemented with some non-deductive procedures. Trivial implications in turn are not directly computed, being less natural, and subject to different types of mistakes. Looking at the linguistic performance of the students in the essays, our research question (RQ7) is whether the information in the tutorial supported by qualitative models was relevant, bringing about modification and improvement in their representation of the interactions between populations. We take the presence of trivial conclusions in the essays to indicate the absence of modification in the representation of the world. Conversely, the presence of non-trivial conclusions should indicate that the information to which the student was exposed was relevant. Some examples from the essays illustrate trivial and non-trivial conclusions, shown in (1) to (4), and in (5) to (8), respectively:

(1) "Hawk eats bird."; (2) "Bird eats spider."; (3) "If man dies, man decrease."; (4) "If hawk is the predator of the bird, the bird is the prey of the hawk."; (5) "Given that the owl population decreased, then the rats increase."; (6) "The aphid population increases because the population of ladybug decreases."; (7) "If spider does not eat ladybug, then bird and hawk decrease."; (8) "If the otter decreases, then fish population increases and alligator and man decrease."

Notice that in (1) and (2) the utterance merely describes a relation between the participants in the food web. We take this description to be old information – which could have been conceptually represented either by means of (previous) formal education or in the course of (informal) everyday life, being part of their knowledge of the world. In (3) and (4), the utterance is an assumption that is rephrased, hence no new information is added. Differently, in (5) to (8), the utterance refers to causal relations between the populations, further representing the dynamics of the food web – the new information that was taught in the tutorial. The manipulation of the causal relation is considered a non-trivial conclusion, which explicates and analyses the content of the input assumptions. Statistical analyses of these show a highly significant reduction in the amount trivial conclusions in the essay produced in Test II, as opposed to the one in the Test I (Mann-Whitney test, $n_1 = 6$; $n_2 = 8$; $U = 7$; $P = 0{,}01$). However, the test showed no significant increase in the amount of non-trivial conclusions. The essays produced in Test II showed that the students clearly employed more elaborated formulations in the linguistic description of the food web. For example, embedded utterances such as "If the fish population increases, the algae decreases, (but) the otter, alligator and man populations increases too" were more frequent in Test II. We noted also that when representing the interaction between predator and prey in written texts, a number of important linguistic questions arise. This interaction involves a bidirectional flow of causality, and the propagation of changes may lead to different results, depending on the starting point (if the predator increases, then the prey decreases; and if the prey decreases, then the predator decreases). The students used a number of different strategies to represent these relations. Among them, some explored verbal tense to define the initial point of the causal flow (e.g. 'population A increases because population B has decreased'). Investigating the mastering of tense on the verbs in (written) Portuguese by deaf students is certainly an interesting topic for future research, given the availability of this encoding in LIBRAS.

5 Conclusions

Making inferences is one of the most important human skills for understanding the world. The study described here showed that the use of qualitative models significantly increased deaf students' ability to make inferences about changes in interacting population. These positive effects were found both in the objective questions and in the written essays the students produced after two tutorial sessions. The students gave, in total, more correct answers to objective questions in Test II than in Test I. An interesting observation was that it is more difficult for the students to recognize propagation of the effects of changes in predators to the prey populations than the contrary. The same difficulty was observed in the written texts, and represents an open issue to be investigated. The study also showed the information in the tutorial supported by qualitative models was relevant, bringing about modification and improvement in their representation of the world (on the interactions between populations). This was confirmed by the observation that the students formulated significantly less trivial conclusions after the use of qualitative models. Finally, this study reinforces our opinion that qualitative models are useful tools to support the educational development of deaf students and the acquisition of Portuguese as second language.

Acknowledgements

We would like to thank the deaf students, their teachers and interpreters *of* LIBRAS, in particular Margot Latt Marinho and Gisele Morrison. Thanks also to Maria Inez Telles Walter for the support with the statistical analyses. Finally, P. and H. Salles are grateful to CAPES / MEC for the financial support to the research project "Portuguese as a second language in the scientific education of deaf".

References

[1] Bessa Machado, V. & Bredeweg, B. (2002) Investigating the Model Building Process with HOMER. In Bredeweg, B. (Editor) *Proceedings of the International workshop on Model-based Systems and Qualitative Reasoning for Intelligent Tutoring Systems*, pages 1-13, San Sebastian, Spain, June 2nd, 2002.

[2] Bouwer, A. and Bredeweg, B. (2001) VISIGARP: Graphical representation of qualitative simulation models. In Moore, J.D.; Luckhardt Redfield, G. & Johnson, J.L. (Editors) *Artificial Intelligence in Education: AI-ED in the Wired and Wireless Future*. Amsterdam, IOS Press, pp. 294-305.

[3] Bredeweg, B. (1992) *Expertise in Qualitative Prediction of Behaviour*. Ph.D. thesis, University of Amsterdam, Amsterdam, The Netherlands, 1992.

[4] Forbus, K.D. (1984) Qualitative process theory. *Artificial Intelligence*, 24:85–168.

[5] Odum, E.P. (1985) Ecologia. Rio de Janeiro, Discos CBS. *Translation of Basic Ecology*, 1983.

[6] Salles, P. & Bredeweg, B.; Araújo, S. & Neto, W. (2003) Qualitative models of interactions between two populations. *AI Communications* 16(4): 291– 308.

[7] Salles, H.; Salles, P. & Bredeweg, B. (2004) Qualitative Reasoning in the Education of Deaf Students: Scientific Education and Acquisition of Portuguese as a Second Language. *In* Forbus, K. & de Kleer, J. (eds.) *Proceedings of the 18th International Workshop on Qualitative Reasoning (QR'04)*, pages 97-104, Evanston, Illinois, August, 2-4, 2004.

[8] Siegel, S. (1975) *Estatística não-paramétrica para as ciências do comportamento*. São Paulo, Ed. McGraw – Hill do Brasil.

[9] Sperber, D. & Wilson, D. (1995) Relevance: Communication and Cognition. Oxford (UK) and Cambridge (Mass), Blackwell Publishers Ltd.

[10] Stevenson, W.J. (1981) Estatística aplicada à administração. São Paulo, Ed. Harper & Row do Brasil.

[11] Weld, D. & de Kleer, J. (eds.) (1990) Readings in Qualitative Reasoning about Physical Systems. San Mateo, CA, Morgan Kaufmann.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Assessing and Scaffolding Collaborative Learning in Online Discussions

Erin Shaw
Center for Advanced Research in Technology for Education
Information Sciences Institute, University of Southern California
4676 Admiralty Way, Marina del Rey, 90292 USA

Abstract: In this paper we present two computational approaches that can be used characterize and measure online threaded discussions and demonstrate that they can objectively validate student-reported differences in collaborative learning between tutor-scaffolded and non-scaffolded discussion activities. The first approach, thread profiling, is used to characterize user interactions that tend to broaden and deepen discussions, and gives insight into how tutors participate in discussions. The second approach, which uses a natural language discourse processor, is used to compare the rhetoric of tutors and students, and shows that tutors consistently use more attributions, elaborations, and enablements to scaffold discussions. To test these ideas we processed twenty-four online activities, constituting over one thousand message posts, during a course at the British Open University. These computational methods and findings have application in virtual tutoring systems and the automated assessment of discussions.

Introduction

Computer mediated communication (CMC) has created an opportunity for social and collaborative learning at a distance. Discussion forums are an integral part of CMC and discussion activities are increasingly co-opted to promote collaborative learning. This presents a problem in that collaborative learning is difficult to characterize, and thus to measure. What techniques do we use, then, to foster collaboration online, and how do we measure their efficacy? We are developing tools to help objectively characterize collaborative learning so we can better assess and scaffold it. Our approach permits the study of corpora of natural text arising from online discussions, complementing ethnographic studies of collaborative discourse.

This work emerged from an online course that featured both tutor-scaffolded and non-scaffolded collaborative discussion activities. Subsequently, students reported that the tutor-scaffolded activities were more collaborative. We used a *Communities of Practice* framework [7] to survey the students and evaluate their perceptions of collaboration, then analyzed twenty-four activities, constituting over a thousand messages, to determine if we could validate the findings. Using thread profiling, we found that there exist canonical profiles of user interactions that tend to broaden and deepen discussions; the approach gives insight into how tutors participate in discussions. Using a new natural language processing tool called SPADE, we compared the rhetoric of tutors and students; the approach confirms that tutors use particular rhetorical relations in greater numbers than do students as a means to scaffold discussions. These real time processing tools can be used to inform how and when virtual tutors might optimally intervene in a discussion to foster collaboration. Both methods can help instructors gain insight into the nature of discussions and are thus valuable tools for the assessment of participation and collaboration.

Communities of practice framework and tutor scaffolding

The theory of Communities of Practice (CoP) [7] posits that learning is a situated, social-cultural activity in which novices move through stages of participation in becoming experts. The theory is based on both the practice of traditional apprenticeship and the social learning theories of Russian historical-cultural psychologists. Although not intended as a pedagogical strategy, CoP involves community, identity, meaning, and practice [22], and is thus an ideal framework for studying *online collaborative learning*, which Clarke and Mayer [3] define as "a structured exchange between two or more participants designed to enhance achievement of the learning objectives." Research on collaborative learning is vast, and automated analysis is playing an increasingly large role. Although we share the goal of characterizing and measuring collaboration, our focus on unstructured discussion text and tutor scaffolding differentiates our work from research on the computational analysis of collaborative activity within global structured environments, such as the DEGREE system and the Collaboration Management Cycle framework [1,2,19].

Thorpe [21] uses the framework of CoP to examine collaborative learning, observing that asynchronous communication has made it possible to foster group work and support it at a distance. She notes Daniel and Marquis' [4] definition of interaction as the archetypal form of learning, their argument that person-to-person interaction is essential, and that interaction and independence are complementary modes of learning. Despite this sentiment, Perraton [16] reports that at the British Open University (UKOU) only half the students participate in online discussion conferences, even when encouraged to do so. Thus, the UKOU employs trained course tutors to encourage, or *scaffold*, interaction among students. Scaffolding is a metaphor for "effective intervention by a peer, adult, or competent person in the learning of another person" [13,23]. Scaffolding is integral to the theory of CoP; here we focus on *explicit* or *intentional* scaffolding by tutors in online discussion activities. Though there have been extensive studies of tutor agency in distance learning communication, most are ethnographic in nature [6,8,9,14,15].

1. Discussion Context and Student Report

The online discussions we report on took place in 2004, during a thirty-week online graduate course on *Understanding Distributed Learning* at the British Open University (UKOU). Twenty-seven students and two tutors participated, with each tutor supervising about half the group. Discussions at the UKOU take place within Open Text Corporation's *FirstClass* conferencing system. All discussions are asynchronous; there is no real time component. There are two types of discussion activities, *Study Guide Activities* (SGAs) and *Tutor Group Activities* (TGAs). SGAs are detailed discussion activities related to readings that are provided by the course study guide. Participation is encouraged but not compulsory. Tutors occasionally contribute resources and reflections. TGAs are structured, two-week learning assignments in which only the small tutor group participates. TGAs are compulsory (graded) and students are expected to spend several hours working on them, and often spend longer. Tutors work closely with students to scaffold learning in TGAs.

We asked students in one tutor group about their own perceptions of collaborative learning as a way to gauge the relevance of the computational findings. Students rated five statements about tutor scaffolding and collaboration, shown in Table 1, on a Likert scale of 1-5 (1= strongly disagree, 5=strongly agree). They also answered questions that placed them in one of three groups, *first timer*, *experienced*, and *experienced and knew a classmate* (e.g., from a previous course). Averages per group are shown in Table 2. Students felt that TGA discussions encouraged collaborative learning, that the tutor was necessary for scaffolding interaction, and that the scaffolded TGAs were more collaborative than the SGAs. This feeling increased for experienced students, and rose to almost 100% for students who were both experienced and knew a classmate in the tutor group. There were no first timers who knew a classmate, although this could occur, especially in an on-campus setting. There appears to be a learning curve associated with online collaborative learning that is influenced by experience and association. We see this especially in the first timers who are mostly neutral about the role of the tutor with respect to scaffolding. This agrees with 'movement toward a center of participation' in the theory of Communities of Practice [7].

Table 1: Survey questions.

Questions	
Q1	I feel that, in general, the tutor was necessary for scaffolding group interaction.
Q2	I feel that, in general, the TGA discussions encouraged collaborative learning.
Q3	I feel that, in general, the TGA discussions were more collaborative than the SGA discussions (please discount the fact that the former were graded)
Q4	Did you 'know' anyone in your tutor group? (Check names, if answer is yes.)
Q5	I have interacted online (participated in collaborative learning activities) in a previous class.

Table 2: Survey answers for all groups.

LS avg	All (12)		First timers (3)		Experienced (9)		Exp & Knew Classmate (4)	
Q1	3.6	8 agree	2.7	1 agree	3.9	7 agree	4.0	4 agree
Q2	3.6	8 agree	3.7	2 agree	3.6	6 agree	4.3	4 agree
Q3	3.5	7 agree	3.7	2 agree	3.4	5 agree	4.0	3 strong agree

2. Interaction Analysis and Thread Profiling

We looked first for patterns of scaffolding and collaborative interaction by analyzing the nature of the discussion threads. Using the *FirstClass*-generated summaries of discussion forums, we filtered and processed the data. We ignored replies-to-self, which tend to be corrections or addendums to original replies, and do not generally contribute to deepening a thread. (General processing details are discussed further in Section 4.) The main idea was to examine the role of tutor and author agency in relation to the breadth and depth of the discussions threads, and to contrast potential differences in agency within the tutor-scaffolded TGAs and the non-scaffolded SGAs.

The analysis of a TGA is shown in Table 3. For each student, we show the number of initial and total posts, and the number of new threads that the initial posts spawned. For each new thread, we show the number of posts in the thread and the number of people participating in the new posts, and the maximum depth and breadth of the new thread. The replies are then analyzed at each depth (D). Replies to the initial post, i.e. "Re: *subject*" occur at depth one (D1), the next level of replies occurs at depth two (D2), and so on.

For example, student A initiated a thread comprising a total of fifteen messages by six participants. The maximum depth and breadth was four and six, respectively. There were four peer replies to the initial post (indicated by 'p') and six replies to the first four, four by peers and two by the author of the initial post (indicated by 'a'). There were four subsequent responses at depth 3 and a final reply at depth 4. The tutor did not participate in this particular discussion, but did participate (indicated by 't') in seven of the eleven discussions.

Similarly, an SGA analysis was performed. Students typically posted only one initial message in an SGA and the tutors usually did not participate. There were fewer instances of author follow-ups to replies than in the TGA. Whatever the reason (e.g., that grades were given for TGAs but not SGAs), the contrast in quality between the TGA and SGA threads validates the students' contention that TGAs were more collaborative.

Table 3. Statistical analysis of a scaffolded tutor group activity (TGA) discussion.

Name	Init Posts	Total Posts	New Thrds	New Thrd Posts/ People	New Thrd Depth/ Breadth	Posts per Depth (D) e.g., D1(n) = n replies to initial post (a = author post, p = peer post, t = tutor post)
A	1	5	1	15/6	4/6	D1(4p) D2(4p,2a) D3(4p) D4(1p)

B	2	3	2	1/1 1/1	1/1
					1/1
						D1(1p)
						D1(1t)

C	1	4	1	4/3	2/2	D1(1p,1t) D2(1p,1t)

D	2	4	2	2/2
				3/2
					1/2
					3/1
						D1(1p,1t)
						D1(1t) D2(1a) D3(1t)

Tutor	0	13	0	0/0	0	
F	1	1	1	2/2	1/2	D1(1p,1t)
G	2	11	1	2/2	3/1	D1(1a) D2(1p) D3(1a)
H	1	7	1	6/4	3/2	D1(1p,1t) D2(1a,1t) D3(1p,1t)

I	3	11	2	2/2 8/4	2/1
					4/3
						D1(1p) D2(1a)
						D1(1p,1t) D2(2a,1t) D3(1p,1t) D4(1a)

The results give insight into how tutors participate in discussions. Of forty seven replies, the tutor intervened a total of thirteen times, or 28% of the time. However, tutor posts produced only four follow-up replies, all from the initial author, indicating that tutor scaffolding was effective only 31% of the time and that scaffolding targeted the individual as opposed to the group. These interventions are consistent with a profile of student-tutor-student interaction, that is, with the student responding to the tutor iteratively.

If we look closely, we notice several common forms. These forms, or *thread profiles,* are characterizations of expected or consistent interaction between two or more individuals. Common profiles include *author-follow-up*, which are instances of author-student-author interaction, *tutor-follow-up*, which are instances of tutor-student-tutor interaction, *student-follow-up*, *broad-shallow*, and *narrow-deep.* Three of these are shown in Figure 1. We see more tutor follow-ups in the TGAs and more author follow-ups in the SGAs, particularly as part of narrow-deep interactions. All of these are potentially collaborative interactions. By looking at these profiles over time and correlating them to student surveys and instructor assessment, we can begin to characterize the nature of collaborative learning.

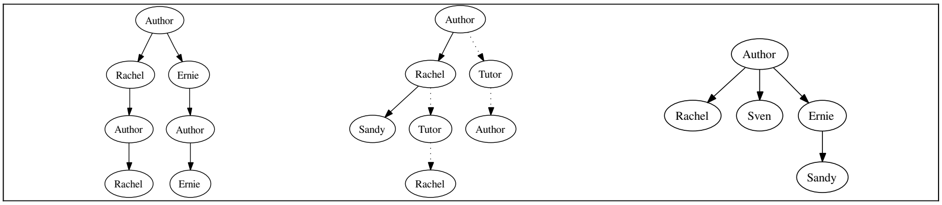


Figure 1. Three canonical interaction profiles found in the data: author follow-up (left), tutor follow-up (middle), and broad-shallow interaction (right).

3. Rhetorical Analysis of Tutor Scaffolding Strategies

What are the strategies tutors use to encourage discussion and, by extension, collaboration, and how might we identify them? Roehler and Cantlon's [17] scaffolding types, shown at right in Figure 2, have been confirmed to be effective in computer mediated learning [13,18]. These types can be mapped to rhetorical relations, at left. *Explanation*, *verification* and *clarification* map to the relations of *elaboration*, *interpretation*, and *restatement*, while *modeling*, *generating* and *inviting* map generally to the presentational relations.

	Rhetorical Relations
S u b j	Elaboration, Relations of Cause, Condition & Otherwise, Interpretation & Evaluation, Restatement & Summary, Sequence & Contrast, Purpose
P	Motivation (increase desire)
r	Antithesis (incr. positive regard)
e	Background (increase ability)
s	Enablement (increase ability)
e	Evidence (increase belief)
n	Justify (increase acceptance)
t	Concession (incr. positive regard)

Scaffolding Types
Offering explanations
Inviting students' participation
Verification and clarification of students' understandings
Modeling of desired behaviors
Generating questions and comments as in think-aloud
Inviting students to contribute actively

Figure 2: Rhetorical relations (Mann & Thompson,) and Scaffolding types (Roehler & Cantlon)

RST discourse parsing

Rhetorical Structure Theory (RST) is a descriptive theory of the organization of natural text[1] that grew out of studies of computational linguistics [11,12]. RST explains the coherence of text in terms of hierarchically-structured rhetorical *relations* that hold between two portions of text. Mann and Thompson [11] suggest a classification of relations based on the effect a relation has on the reader. '*Subject matter relations* are those whose intended effect is that the reader recognize the relation in question*; presentational relations* are those whose intended effect is to increase some inclination in the reader, such as the desire to act or degrees of positive regard for, belief in, or acceptance of the nucleus.' We used an RST analysis of discussions to validate student reports that tutors helped scaffold discussions.

SPADE (Sentence-Level Parsing of Discourse) is an RST discourse parser that purportedly achieves near-human levels of performance (defined as 90% accuracy) in the task of deriving sentence-level discourse trees [20]. A SPADE parse of a tutor's post is shown in Figure 3. Three relations generally stand out in tutor messages: *attribution* (the writer wants to make the owner of the text clear to the reader), *elaboration* (the writer wants to make it easier for the reader to understand), and *enablement* (whereby the writer wants to increase the potential ability of the reader). Other relations that occur regularly in all messages include *background*, *cause*, *comparison*, *condition*, *contrast*, and *explanation*.

Rhetorical analysis of all TGA and SGA discussions

To confirm that the tutor-student differences were true in general we looked at the discussions that took place during fourteen scaffolded and ten non-scaffolded activities and compared general tutor use of relations to general student use. We used percentages to normalize the results, so that an *attribution* value of 4.9 indicates that the participants used

[1] The textual representation of natural language.

```
(Root (leaf 1)
(text _!Hi everyone_!))
(Root (span 1 2)
  ( Nucleus (leaf 1) (rel2par span)
(text _!The attached file contains the
instructions for the TGA1 activity_!) )
  ( Satellite (leaf 2) (rel2par Elaboration)
(text _!which begins today ._!) ))
(Root (span 1 2)
  ( Satellite (leaf 1) (rel2par Attribution)
(text _!Please ensure_!) )
  ( Nucleus (leaf 2) (rel2par span)
(text _!that all your responses to this activity
are placed in this conference ._!) ))
(Root (span 1 6)
  ( Satellite (leaf 1) (rel2par Attribution)
(text _!Do remember_!) )
( Nucleus (span 2 6) (rel2par span)
( Nucleus (span 2 3) (rel2par span)
     ( Nucleus (leaf 2) (rel2par span)
(text _!that I am here_!) )
     ( Satellite (leaf 3) (rel2par Enablement)
(text _!to support and encourage you ,_!) ))
    ( Satellite (span 4 6) (rel2par Condition)
     ( Nucleus (leaf 4) (rel2par span)
(text _!so if you have any questions about this
TGA_!) )
     ( Satellite (span 5 6) (rel2par Elaboration)
       ( Satellite (leaf 5) (rel2par Attribution)
(text _!at all please ask me – that's_!) )
       ( Nucleus (leaf 6) (rel2par span)
(text _!what I'm here for : - )_!) )
       ) )
) )
```

Figure 3. SPADE discourse parser output of a tutor's post. The original text is shown in bold lettering.

attributions 4.9% of the time. Table 6 shows the results for the TGAs. Relative to the student group, the tutors' far greater use of attribution, elaboration and enablement relations is evident in both activities. Some of these comparisons are not surprising. For example, we expect the tutor to elaborate to a greater degree since these posts include instructions. Others are noteworthy: The tutor provided deeper explanations – resulting in more background, causations, comparison, contrasts, and conditions, as well as a higher number of attributions - and used more enablements to increase the ability of their students. Joint relations are indicated by conjunctive clauses, and are perhaps a result of deep explanations.

Table 7 shows the results for the SGAs. Two things are worth noting: There is an increase in the tutors' use of attribution and elaboration, and the tutors use comparison and explanation (whereas they do not in the TGAs). This may be because original quotes were included in the messages, so that the comparisons and explanations may be the students' (see Section 4, below) or it may correspond to the tutors' contributions of resources and reflections in the SGAs, as opposed to what they consider traditional scaffolding strategies in the more structured TGAs.

Table 6. Tutor-scaffolded activities: Rhetorical relations as a percent of messages posted.

Tutor-scaffolded Activities (TGAs)										
Role/%	#Msg	Attrib	Bkgrd	Cause	Comp	Cond	Contr	Elabor	Enabl	Expl	Joint
Tutor	172	4.9	1.2	0.2	0.1	0.8	0.6	10.6	1.6	0.1	5.9
Student	492	3.0	0.6	0.1	0.0	0.4	0.3	5.5	0.7	0.0	4.2

Table 7. Non-scaffolded activities: Rhetorical relations as a percent of messages posted.

Non-scaffolded Activities (SGAs)											
Role/%	#Msg	Attrib	Bkgrd	Cause	Comp	Cond	Contr	Elabor	Enabl	Expl	Joint
Tutor	26	26.5	1.5	0.0	3.9	0.0	0.0	51.15	4.23	1.92	25.4
Student	401	6.8	1.6	0.4	0.3	0.8	1.1	10.7	1.6	0.2	8.9

Combining approaches for better understanding

Rhetorical analysis and thread profiling might be combined to show how different rhetorical patterns, both tutors' and students', influence interaction profiles; for example, to deepen or broaden discussions in both scaffolded and non-scaffolded activities. We might analyze messages that elicit many responses, or different profiles of responses; or investigate gender differences in scaffolding strategies. These findings can be used to aid both human and machine tutors who wish to improve their scaffolding techniques, and the characterizations of collaboration that emerge, if validated by participants, can be used to evaluate learning in discussion forums.

4. Processing Discussion Data

We encountered several pitfalls in processing the data; two are unique to the *FirstClass* conferencing system. *FirstClass* replies are inferred by a *Re()* in the subject line, instead of by using a unique message or thread identifier. If the subject line is changed manually, it may be identified incorrectly, and if there are multiple replies to message posts with identical subject lines, there may be no way to automatically untangle the threads. The latter problem was the case with many of the SGA discussions and the threads had to be differentiated manually. Differentiating threads was straightforward when original quotations from the previous post were included in the reply, which might help automatically differentiate threads, but presented a problem for SPADE processing because *FirstClass* identifies the start of the quotation (by including a line, e.g., "Erin Shaw writes:" before the quote), but not the body or close of one. (Many text editors make it easy to identify quotes by preceding these lines with a "greater than" symbol (">").) In addition, SPADE requires that posts be marked up for processing; however, messages occasionally contain malformed URLs and other incoherent text that preclude successful processing. A final general pitfall is in how attachments are used. In a few of the activities, some students included an attachment with their answers while others did not, and this inconsistency was not taken into account.

5. Conclusion

The task of assessing collaborative learning in online discussions is difficult, and most studies to date have been qualitative in nature. In this paper, we have shown that computational tools for analyzing corpora of threaded discussions can be applied to the difficult task of characterizing, measuring and scaffolding collaboration. A basic research approach has been taken; though preliminary, the results show that computational analyses support student findings that some discussions are more collaborative than others and that tutor scaffolding plays a role in collaboration, even while 'collaboration' is an elusive term. Using thread profiling, we found that there exist canonical profiles of user interactions that give insight into how tutors participate in discussions. Using a new natural language processing tool to compare the rhetoric of tutors and students, we confirmed that tutors use particular rhetorical relations in greater numbers than do students as a means to scaffold discussions. We envision that combining the approaches, especially within a CoP framework, will produce many interesting and detailed characterizations that will help produce metrics for measuring collaboration and the efficacy of techniques to scaffold it.

6. Acknowledgements

The work was supported in part by a grant from the Lord Corporation Foundation to the Distance Education Network at the University of Southern California Viterbi School of Engineering. The author thanks Karelena Mackinlay and the students of UKOU H805 for their assistance with the study.

References

[1] Barros, B., Mizoguchi, R., & Verdejo, M. A platform for collaboration analysis in CSCL. An ontological approach. Available at http://www.ei.sanken.osaka-u.ac.jp/pub/miz/BarMizVerPoster.pdf [9/15/04]

[2] Barros, B. & Verdejo, M. (2000) Analysing student interaction processes in order to improve collaboration. International Journal of Artificial Intelligence in Education, 11.

[3] Clark & Mayer, e-Learning and the Science of Instruction (2003), Pfieffer.

[4] Daniel, J., & Marquis, C. (1979). Interaction and independence: Getting the mixture right. *Teaching at a Distance, 15*, 25-44.

[6] Hara, N., Bonk, C.J., Angeli, C. (1998) Content Analysis of Online Discussion in an Applied Education Psychology Course, CRLT Technical Report No. 2-98, Kluwer Academic Publishers, the Netherlands. Republished with permission from Instructional Science, 28:2 pp. 115-152, 2000.

[7] Lave, J. and Wenger, E. (2001) 'Legitimate peripheral participation in communities of practice', in Lea, M.R. & Nicoll, K. (eds.) Distributed Learning: Social and cultural approaches to practice, pp. 56-63.

[8] Lea, M. (2001) Computer Conferencing and Assessment: new ways of writing in higher education, Studies in Higher Education, Volume 26, No. 2.

[9] Light, V. & Light, P. (1999) 'Analysing asynchronous learning interactions: computer-mediated communication in a conventional undergraduate setting, in: K.Littleton & P. Light (Eds) *Learning with Computers; analyzing productive interactions*, pp. 162-178 (London, Routledge)

[10] Light, V., Nesbitt, E., Light, P., & Burns, J.R. (2000) 'Let's You and Me Have a Little Discussion': computer mediated communication in support of campus-based university courses. *Studies in Higher Education* Volume 25, No. 1.

[11] Mann, W.C. and Thompson, S.A. (1987) Rhetorical Structure Theory: A Theory of Text Organization, University of Southern California, Information Sciences Institute report number ISI/RS-87-190, NTIS ADA 183038. Available at http://www.sil.org/%7Emannb/rst/documentaccess.htm [Accessed 9/21/04]

[12] Mann, W. (1999) An Introduction to Rhetorical Structure Theory (RST). At http://www.sil.org/%7Emannb/rst/rintro99.htm [Accessed 9/21/04]

[13] McLoughlin, C. (2002) Learner support in Distance and Networked Learning Environments: Ten Dimensions for Successful Design. *Distance Education*, Vol 23, No 2

[14] Ng, K.C. (2001) Using E-mail to Foster Collaboration in Distance Education, Open Learning, Vol. 16, No. 2.

[15] Painter, C., Coffin, C. & Hewings, A. (2003) Impacts of Directed Tutorial Activities in Computer Conferencing: A Case Study. Distance Education, Vol. 24, No. 2.

[16] Perraton, H. (2000) *Open and distance learning in the developing world*, London , Routledge.

[17] Roehler, L., & Cantlon, D. (1997). Scaffolding: A powerful tool in social constructivist classrooms. In K. Hogan & M. Pressley (Eds.), *Scaffolding student learning: Instructional approaches and issues.* Cambridge, MA: Brookline.

[18] Salmon, G. (2001). *E-moderating: The key to teaching and learning online*. London: Kogan Page.

[19] Soller, A., Jermann, P, Muhlenbrock, M, Martinex, A. (2004) Designing Computational Models of Collaborative Learning Interaction: Introduction to the Workshop Proceedings. In Proceedings of the 2nd Inter'l Workshop on Designing Computational Models of Collaborative Learning Interaction, ITS 2004.

[20] Soricut, R. and Marcu, D. (2003). Sentence Level Discourse Parsing using Syntactic and Lexical Information. Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL), May 27-June 1, Edmonton, Canada.

[21] Thorpe, M. (2001) 'From independent learning to collaborative learning: New communities of practice in open, distance and distributed learning", in Lea, M.R. & Nicoll, K. (eds.) Distributed Learning: Social and cultural approaches to practice, pp. 131-151.

[22] Wenger, (1998). *Communities of practice: Learning, meaning, and identity.* Cambridge, MA: Cambridge University Press.

[23] Wood, D., Bruner, J. S. & Ross, G. (1976). The role of tutoring in problem solving. *Journal of child Psychology and Psychiatry,* 17(2), 89-100.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

THESPIAN: An Architecture for Interactive Pedagogical Drama

Mei Si, Stacy C. Marsella and David V. Pynadath
Center for Advanced Research in Technology for Education
USC Information Sciences Institute
4676 Admiralty Way, Marina del Rey, CA, 90292 USA
{mei,marsella,pynadath}@isi.edu

Abstract. Interactive drama is increasingly being used as a pedagogical tool in a wide variety of computer-based learning environments. However, the effort required to build interactive dramas is quite significant. We built Thespian, an architecture that supports faster development of IPDs, open-ended interaction, encoding of pedagogical goals and quantitative metrics for evaluating those goals. Thespian uses autonomous agents to control each character and assumes that the starting point for the design process is a set of standard scripts. A "fitting" algorithm facilitates the design process by automatically adjusting the goals of the agents so that the agents perform their roles according to the scripts. This also ensures the agents will behave true to their character's motivations even when the interactive drama deviates from the scripts. In this paper, we discuss this basic approach in detail and illustrate its application to the Tactical Language Training System.

Keywords. pedagogical agents, authoring & assessments tools, language learning

1. Introduction

Interactive drama is increasingly being used as a pedagogical tool in a wide variety of computer-based learning environments (e.g., [5,6,9,10]). In an interactive pedagogical drama (IPD), the learner interacts with the characters in a story and the story unfolds based on those interactions. Ideally, an IPD combines the pedagogical power of drama with a more active learning experience that allows learners to explore a simulated story world and see the effect of their actions. The engaging nature of drama and the direct link between actions and outcomes ideally engages students more, motivates them to spend more time learning (e.g., to explore possible paths in the story), and appropriately contextualizes the experience.

However, the creation of interactive pedagogical drama faces several challenges. Up to now, the effort required to design and build interactive dramas is quite significant [3,8], potentially requiring man-years of design and implementation. Further, effective design for relatively open-ended user interactivity is still an open research issue. And there is often a tension between the goal of interactivity and the goal of creating an engaging drama with consistent, well-motivated characters. Satisfying both goals can be a significant technological and creative challenge. For example, writers often do not have expertise in designing interactive stories, which is still largely a nascent art form. More fundamentally, in an interactive pedagogical drama, pedagogical goals must also be achieved. This raises the question of how the pedagogy is embedded in the environment, how the experience of playing the game leads to desired learning outcomes and what the metrics are that determine whether pedagogical goals have been achieved.

We have developed an approach that speeds up the development of IPDs, supports open-ended interaction, achieves pedagogical and dramatic goals and supports quantitative metrics for evaluating the learner's achievement. We call our system Thespian, due to its actor-centric approach to realizing IPDs. Thespian's basic architecture uses autonomous software agents to control each character, with the character's personality and motivations encoded as agent goals. The ability of goal-driven agents to autonomously select actions based on the current state of the world allows them to be responsive to open-ended user interactions, while staying consistent with their "personality". We ensure that the learner's experience in the drama is consistent with pedagogical goals by embedding them in the drama; the world and characters in the world behave in ways that reinforce the lessons that the IPD is trying to teach. We can then define quantitative metrics on the achievement of pedagogical goals in terms of what happens in the story.

Thespian assumes that the starting point for the design process is a standard script or story outline, with possible variations, produced by a writer. This approach is typically used (e.g., [3]) because it provides a good baseline for creating an experience that can satisfy dramatic and pedagogical goals. The problem is that going from such linear script material to an interactive agent-based system is an arduous, time-consuming process requiring extensive software skills. We significantly facilitate the process by using an automated "fitting" algorithm [11] that adjusts agents' goals so that they are motivated to perform their roles according to the scripts. This ensures that the agents' autonomous behavior can follow the script when the learner's behavior is consistent with it, but is still true to their character's motivations even when the drama deviates from the script.

In this paper, we discuss this basic approach in detail. We also illustrate its application to the Tactical Language Training System (TLTS) [4] for rapidly teaching students the rudiments of a foreign language and culture.

2. Example Domain: Tactical Language Training System

TLTS is comprised of two main components that mutually reinforce the learning experience: a Mission Skill Builder (MSB), and a Mission Practice Environment (MPE). We will limit our discussion here to the MPE, a 3D role-playing interactive drama for learners to practice using their language skills. In the drama, the learner takes on the role of an army sergeant assigned to conduct a civil affairs mission in a foreign town. The learner navigates in the virtual world and interacts with virtual characters using spoken Arabic and gestures. The MPE contains several training scenes, each requiring the learner to carry out specific tasks within the interactive drama. We focus on the first scene to illustrate our approach to building an IPD. The story begins in a village café. The learner's mission is to establish rapport with the local people and find out about their leader. The learner enters the café and interacts with several of the locals, including an old man and

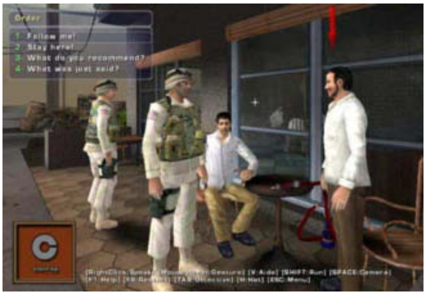


Figure 1.Scene 1 from the MPE.

a young man. The difficulty of the mission varies according to the learner's language skills. In the novice level, both of the locals are relatively cooperative, while in the expert level, the young man worries more about the safety of the town than being helpful. He may accuse the learner of being a CIA agent if he fails to establish trust. If, on the other hand, the learner uses culturally appropriate behavior, the old man will assist them.

3. Desiderata for IPDs

The design of an interactive pedagogical drama must address several requirements. Foremost, the characters should be well-motivated. They should behave according to the scripts when the learner's behavior is consistent with it. Additionally, when there are deviations from the scripts, the characters should behave consistently with the motivations implied by their on-script behavior.

Most critically, an IPD needs to encode the pedagogy. There must be some way for the learner to interact with the system and learn on the basis of that interaction. In particular, a system that has an explicit awareness of its own pedagogical goals is better equipped to work toward them in the face of unexpected interactions with the learner.

Characters must also support and maintain the interaction with the learner. Any dialog-based interactive drama must support social interaction in the form of a dialog with the learner. In practice, this means that characters should follow social norms, unless strongly motivated to violate them. By following norms, characters behave consistent with expectations and provide an incentive to interact with them.

Finally, from a drama designer's perspective, the design process should be as free of technical burdens as possible and ideally support reuse of previously developed materials.

4. Thespian

We developed Thespian as a multiagent system for controlling virtual characters in an IPD. Thespian builds on top of PsychSim, a multi-agent system [7] that controls the characters. PsychSim provides a framework for goal-driven, social behavior that forms a sound basis for meeting the requirements of IPDs that we discussed in Section 3. PsychSim agents generate their behavior through a bounded planning process that seeks to achieve their goals. Thus, the agents will choose only those behaviors that are consistent with their character profiles. PsychSim agents have a "Theory of Mind" that allows them to form mental models about other entities in the IPD, including the learner. Thus, we can potentially encode pedagogical goals as desired conditions on our model of the student. These mental models also allow a PsychSim agent to reason about the effects of its behavior on its relationships with other entities. This social reasoning capability can encode the social norms that support and maintain interactions with the user. Finally, PsychSim provides algorithms for tuning model parameters in response to the desired agent behavior. We can apply such algorithms to simplify the authoring process by ensuring that characters behave according to the script when the learner's behavior is consistent with it. This section describes how we built Thespian on top of these basic capabilities.

4.1. Goal-Driven Behavior

PsychSim represents goals as degrees of achievement with regard to certain state features (physical features, relationships, knowledge, etc.). The agents make decisions on what action to perform or what to say based on their beliefs on the possible effects of such decisions. Actions change the physical world in some fixed (possibly uncertain) way. Saying something to another agent changes the beliefs of that agent and of any other agent that may overhear. The agents project into the future to evaluate the effect of each option on the state and beliefs of the other entities in the IPD. The agents consider not just the immediate effect, but also the expected responses of the other entities and, in turn, the effects of those responses. The agent evaluates the overall effect with respect to its goals and then chooses the action that has the highest expected value. From a decision-theoretic viewpoint, we can view this decision procedure as a boundedly rational variation on the standard solution of a *partially observable Markov decision problem (POMDP)* [13]. Thus, every action chosen by an agent is motivated by its goals, although irrational behavior may still arise due to erroneous beliefs.

We use PsychSim's basic goal representation to encode the many possible goals that our Thespian agents may have. We draw from a goal taxonomy from the psychological literature [2]. Many of these goals will conflict with each other in everyday situations. The standard "achievement" goals of logical representations are insufficient to resolve such conflicts because of the ambiguity that arises. PsychSim's decision-theoretic representation allows Thespian to model different character profiles by varying an explicit relative priority among the set of possible goals. Thus, Thespian models a character profile as its various goals and their relative importance (weight). For example, in the MPE, the old man has goals of maximizing its safety level and maximizing the level of being likable, with the latter being weighted as more important. Varying these relative weights leads to changes in the agent's behavior, giving us a wide range of possible characters that will all still act in a consistent fashion with respect to their individual goals.

4.2. Pedagogical Goals

In addition to goals that represent the character profile, our goal representation can encode the degree to which the pedagogy has been successful. We currently envision three approaches to encoding learning goals into Thespian.

First, learning goals can be embedded in the world's dynamics and the characters' goals. For example, one of the pedagogical goals in the MPE is for the student to learn to establish a relationship with the local people, in particular that they trust him/her. We can encode this pedagogical goal into the dynamics by ensuring that failure to establish trust will have consequences. At its most severe, distrust can cause irreparable breakdowns in the relationship. Specifically, in the MPE, if a student fails to achieve even the minimal requirement for this trust goal, the young man will accuse him of being a CIA agent, and all characters will refuse to talk to him. Such breakdowns are one extreme. Characters can also act in ways that help the student. In the MPE, the old man has the goal that he trust the learner, that he feel safe around him, and at times he deliberately behaves in a fashion that would elicit behavior from the student that increases trust. Specifically, the old man can ask the student questions about the student and his mission, which provides more information and makes the old man feel safer. Although it is not an explicit intention of the character, its behavior does assist the learner.

However, Thespian can provide characters with the explicit intention that the student learn. In this approach to encoding the pedagogy, characters have a goal that the learner acquire skills specified by the pedagogy. A character could then use its mental model of the learner as a student model to measure the degree to which the pedagogical goals are achieved. The theory of mind embedded within Thespian forms a subjective view of the world that includes beliefs about the students' knowledge and capabilities based on their behavior. The old man, for example, could have the explicit goal that the student give a high goal priority of establishing trust. Having encoded such a goal, the old man could now evaluate a possible action choice using its mental model of the student's goals to assess the effect on the student and, in turn, on the pedagogical goals so encoded. Again, because we have priorities on the goals, we can choose how much a particular character is driven by pedagogical goals for the learner in relation to its own personal goals.

Finally, a third way to encode the pedagogy is to have a behind-the-scenes director agent that is directing the drama in pedagogically appropriate ways. In other words, we could go even further by explicitly encoding the intention to teach in the overall system through this director agent. The MPE does not employ this technique currently but it is feasible within Thespian. These three approaches to encoding pedagogy (in the world's dynamics, in the character's intentions and in the system's intention) provide Thespian with a rich framework for realizing pedagogical drama.

4.3. Social Norms

While Thespian's ability to encode pedagogical goals gives the agents incentive to exercise the pedagogy, we also must give the *student* the same incentive. One of the motivations underlying IPDs is that the student's inherent social desires can provide an incentive for following the pedagogy if the characters are socially interesting entities. As described in Section 3, characters that are sensitive to social norms can provide such an incentive.

PsychSim provides a general framework for representing states, actions, and the dynamics of the world. While such probabilistic models have typically been used in modeling physical systems, Thespian uses them to model *social* dynamics as well. We constructed Thespian's model of social dynamics by first identifying critical social variables. We have begun by encoding the trust and liking relationships that exist between entities. We then defined domain-independent dynamics for these social variables (e.g., increase your liking of another agent if it does something that helps achieve your goals). Giving an agent goals on these social variables will give it incentive to be liked and trusted by the student. We are currently applying this same methodology to expand our set of social variables to include other key features (e.g., affinity, freedom).

In addition to these more persistent relationship variables, Thespian also uses social variables to represent more temporary obligations that may exist between agents. In general, actions by one agent can impose a type of obligation on another, and a certain set of responding actions will satisfy the obligation to some degree. We currently use these obligations to encode a broad set of social norms as pairs of initiating and responding actions: greeting and greeting back, introducing oneself and introducing oneself back, conveying information and acknowledging, inquiring and informing, thanking and saying you are welcome, offering and accepting/rejecting, requesting and accepting/rejecting, etc. For example, Thespian's dynamics for "inquiry" specify that one of its effects is the establishment of an obligation on the part of the inquiree to satisfy the enquirer (e.g., by providing the needed information).

By giving the agents goals to satisfy any such outstanding obligations, we give them an incentive to follow the encoded social norms. In some cases, the agents may already have an incentive from relationship goals in addition to the obligational ones. For example, an agent providing information in response to an inquiry will be helping the enquirer achieve its goals, leading to a stronger liking relationship. Alternatively, social norm goals may conflict with the agent's other goals, leading to possible violations. For example, an agent may decide not to satisfy an inquiry obligation, because revealing the requested information may reveal vulnerabilities, threatening the security of the agent. The relative priorities among all of these goals reflect the value that the character places on the corresponding social norms. These values are often culturally specific and can also vary according to its personality. However, although we vary the relative weights on the norms from character to character, the underlying mechanism for representing and maintaining norms and obligations does not change, so we can reuse it across many IPDs.

4.4. Authoring

We have shown how Thespian encodes personalities, pedagogical goals, and social behaviors as goals that can drive autonomous agent behavior. Because of this autonomy, the author of the IPD no longer has to specify all of the possible behaviors of the character. However, the character's behavior now depends on the goal priorities chosen by the author, so we simply replaced the previous authoring task with a new one. Furthermore, the process of tuning such quantitative parameters is typically less natural to the author than writing a script.

Fortunately, PsychSim provides an algorithm for automatically choosing these goal priorities based on a few instances of desired behavior [11]. Thespian uses this algorithm to take *partial* scripts, provided by the author, and automatically tune the relative goal weights among the personal, pedagogical, and social goals of the character. Once Thespian has fit the character's goals to this input, the character will always generate autonomous behavior that is consistent with the given scripts, when applicable. Furthermore, when the learner's interactions lead them off the scripts, the agent will still act consistently with its goals. In other words, the fitting process extrapolates from the partial scripts to an exhaustive specification of consistent behavior over all situations. It is as if we were "teaching" the agent the motivations of its character, as opposed to having them simply memorize the scripts.

Thespian reduces authoring effort in two ways. First, Thespian's authoring process alleviates burden on authors by not requiring them to craft all possible paths through the story, while still allowing a more natural process than required by hand-tuning parameters. Second, Thespian supports the reuse of characters and environments across IPDs. Thespian can separate the models of characters from those of the environment they are in. Dynamics designed for one IPD environment can be reused in another. And after fitting, an agent becomes a character with a certain set of goals. This character can be easily plugged into other stories to play a similar role. See [12] for further discussion.

5. Results and Current Status

Figure 2 provides an excerpt of actual dialog between the (human) student, the student's aide and two locals from scene one. Note Figure 2 shows the surface language form, but this form is mapped by the speech recognizer to an internal speech act representation that the agents can reason about. Prior to this excerpt, the student has only introduced his name, but has not introduced the aide or details about their mission. The impact of having failed to make a proper introduction is that he has not built trust with the locals.

In the first line of Figure 2, the student asks the old man a sensitive question. However, the young man then seizes the dialog turn because he perceives a potential security threat by someone he does not trust. Through his mental model of the old man and his

Speaker	Addressee	Utterance
Student	Old man	*minu mas'uul b-hel-manTaqa?* Who is the most important official in this town?
Young man	Old man	*9ala keefak!* Slow down!
Young man	Student	*u hiyye minu?* Who is *she*? (referring to aide)
Old man	Student	(silence)

Figure 2. Excerpt from Scene 1 dialog.

lookahead reasoning, the young man can foresee that if he does not stop the old man, the old man would give the answer to the student, which would hurt his own goal of safety. If he instead asks the student a question, he can not only stop the old man from giving the answer, but also gain safety by getting more information from the student. This reasoning leads the young man to tell the old man not to answer the question (second line from Figure 2) and to ask who the aide is (third line from Figure 2). The young man has both the goal of increasing safety and following social norms. According to the latter goal, he should keep quiet, because the student is asking the old man a question but he values safety more than following social norms. So, in this case, he picks the action that increases his safety, even if it violates social norms. For the old man, following social norms is the most important goal. He has two obligations. The student's question to him imposes an obligation to answer. The young man's question to the student imposes an obligation for the old man to keep quiet (i.e., wait for them to finish their conversation). The more recent obligation receives higher priority. Therefore, he chooses to keep silent.

This rather complex exchange was achieved by the automated fitting process. Fitting adjusted the characters' goal weights (of safety and following social norms) to achieve the behaviors exhibited in this example.

Currently the MPE has three scenes. These scenes have as many as six characters plus the student's character. All three scenes are constructed by using automated fitting. The TLTS system has so far undergone six stages of formative evaluation during the development process. We got mostly positive feedback about its effectiveness for language training. Since April 2004 to January 2005, we have gone through three rounds of testing with a total of 30 subjects. So far, the overall evaluation of the MPE is that it is successful in providing an engaging environment, and is an effective assessment tool [1]. Beginning in March of this year, we will have another round of testing with at least 100 subjects.

6. Conclusion

The promise of interactive pedagogical drama has often been thwarted by the arduous design and programming tasks facing the creators of such systems. Thespian facilitates the design process of agent-based IPDs in several ways. It enlists automation in the character configuration process to simplify authoring. It also provides multiple ways to support pedagogical goals. Additionally, Thespian provides a methodology for modeling social dynamics within a decision-theoretic framework.

Thespian simplifies the authoring process in several ways. Agents are motivated solely by their goals and their goals are automatically fitted so that they perform according to the scripts. Because their behavior is driven by their goals and not simply scripted, the agents respond to unexpected user interaction in ways consistent with their motivations. If they do not, the misbehavior can also be fed into the fitting process. In the MPE, we have demonstrated how to embed pedagogical goals in the dynamics of the story world and have discussed additional approaches. We believe these techniques can be applied to other IPDs as well.

Going forward, the vision of Thespian would be for non-technical designers to author dramas on their own. There are still steps in the process that are impediments to such a vision, including translating scripted dialog into the formal speech act language that the agents understand. We plan on addressing such impediments in our future work.

Acknowledgments

This project is part of the DARWARS Training Superiority Program of the Defense Advanced Research Projects Agency. The authors wish to acknowledge the contributions of the members of the Tactical Language Team.

References

[1] C. Beal, W.L. Johnson, R. Dabrowski, and S. Wu. Individualized feedback and simulation-based practice in the tactical language training system: An experimental evaluation. In *AIED*, 2005.

[2] A. Chulef, S.J. Read, and D.A. Walsh. A hierarchical taxonomy of human goals. *Motivation and Emotion*, 25:191–232, 2001.

[3] W. Swartout et al. Toward the holodeck: Integrating graphics, sound, character and story. In *Agents*, pages 409–416, 2001.

[4] W.L. Johnson et al. Tactical Language Training System: An interim report. In *Proc. of the Internat'l Conf. on Intelligent Tutoring Sys.*, pages 336–345, 2004.

[5] I. Machado, A. Paiva, and P. Brna. Real characters in virtual stories: Promoting interactive story-creation activities. In *Proc. of the Internat'l Conf. on Virtual Storytelling*, 2001.

[6] S. Marsella, W.L. Johnson, and C. LaBore. Interactive pedagogical drama for health interventions. In *Proc. of the Internat'l Conf. on Artificial Intelligence in Education*, 2003.

[7] S. Marsella, D.V. Pynadath, and S.J. Read. PsychSim: Agent-based modeling of social interactions and influence. In *Proc. of the Internat'l Conf. on Cognitive Modeling*, pages 243–248, 2004.

[8] M. Mateas and A. Stern. Integrating plot, character and natural language processing in the interactive drama Fa cade. In *Proc. of the Internat'l Conf. on Tech. for Interactive Digital Storytelling and Entertainment*, 2003.

[9] A. Paiva, J. Dias, D. Sobral, R. Aylett, P. Sobreperez, S. Woods, C. Zoll, and L. Hall. Caring for agents and agents that care: Building empathic relations with synthetic agents. In *Proc. of the Internat'l Conf. on Autonomous Agents and Multiagent Systems*, pages 194–201, New York, 2004. ACM Press.

[10] L. Plowman, R. Luckin, Laurillard, M. Stratfold, and J. Taylor. Designing multimedia for learning: Narrative guidance and narrative construction. In *CHI*, pages 310–317, 1999.

[11] D.V. Pynadath and S. Marsella. Fitting and compilation of multiagent models through piecewise linear functions. In *AAMAS*, pages 1197–1204, 2004.

[12] M. Si, S. Marsella, and D.V. Pynadath. Thespian: Using multi-agent fitting to craft interactive drama. In *AAMAS*, 2005.

[13] R.D. Smallwood and E.J. Sondik. The optimal control of partially observable Markov processes over a finite horizon. *Operations Research*, 21:1071–1088, 1973.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Technology at work to mediate collaborative scientific enquiry in the field

Hilary SMITH*, Rose LUCKIN, Geraldine FITZPATRICK*,
Katerina AVRAMIDES, Joshua UNDERWOOD
Interact Lab, IDEAS Lab*
Human Centred Technology Group
Department of Informatics, University of Sussex
Brighton, BN1 9QH, UK

Abstract. This paper describes and contrasts findings from two related projects where groups of science pupils investigated local air pollution using a collection of mobile sensors and devices. Both projects however played out in different ways. A qualitative analysis of the projects points to the various issues that contributed to the different experiences despite similar technologies for a similar task. These include: project focus; type of facilitator input and the benefits of in-situ data collection combined with subsequent review and reflection. We point to specific relationships between technologies and context of use, and building on this draw out recommendations for the design of in-context, science learning sessions. This work contributes to the growing conceptual understanding, based on 'real world' experiences, of how mobile and ubiquitous technologies can be appropriated in context to support learning. It contributes to an increased understanding of the types of collaborative scientific activity that are supported by different technology configurations, and the roles that human and system facilitators can play in this process.

1. Introduction

Wireless, mobile and ubiquitous technologies are generating a profusion of potential new ways to engage a generation of inquisitive, technology-savvy students [3, 6]. Combined with exploratory styles of learning, they could support a variety of activities employed by teachers in inspirational, novel and real world learning situations [e.g. 8]. While this potential is widely acknowledged, the question of how best to apply these technologies in learning contexts is still open for discussion and exploration, with relevant concepts, theories and guidelines only starting to emerge. We compare and contrast two studies in which sensing technology was used to afford learners a combination of automatic and manual data collection in two different locations. In this way we can take good account of the contextual factors (e.g. in-situ data collection, type of facilitator input) that influence the ways in which learners and devices interact and also abstract away from the specifics of any single context to contribute to a more general understanding about how we might best use and integrate devices into learning tasks and contexts.

Specifically, we report on two related projects that explore issues around public understanding of e-Science, mobile technologies and learning. We used an exploratory research approach to understand the potential of mobile devices when used as part of a collaborative data-collection process. The emphasis of the first (e-Science) project was on a loosely structured, technology rich session with young students collecting pollution data on a university campus. The second (SENSE) project focussed on a complete scientific enquiry lifecycle, where students explored pollution in their school locality. Both projects used the same suite of data gathering devices, e.g. a Carbon Monoxide (CO) monitor and both had small teams of young scientists working with an adult facilitator.

Our interest here is that even though they utilised the same equipment, the data-logging sessions within each study differed in the level of structure, whether they were single sessions or part of a series of sessions, the role of the facilitator, the type of device/task given. Through qualitative comparative analysis of video and log data, we identify the main issues that arose during the different data collection sessions (task focus, pre-session activities, device control, review activity). These issues point to the ways that the contextual factors contribute to the appropriation of similar tools for similar tasks in different ways. Based on this, we report recommendations for the design of technologies and their use in the educational field-work setting. We conclude by identifying opportunities for focussed studies in this area.

2. Related work and theoretical grounding

Support for collaboration and communication across time and space represent key potential benefits that should be gained from the development of mobile and ubiquitous technologies. These technologies should also allow learners from the nursery to university and beyond to access resources, such as information, software and experts or more knowledgeable peers, to enrich their educational experience and increase their understanding. However, to make the most of what this technology has to offer we need to understand the contextual and social as well as the cognitive (and meta cognitive) aspects of the learner's experience. We have seen that a hands-on experience can lead children to be more imaginative in their understanding of the inter-workings of a living woodland [7]. Both motivational and cognitive benefits have been found when students have greater ownership of their data through data-logging (e.g. see [9]: students learn to focus more on content than the logistics of manual data capture, thus freeing them to interpret and theorise what the data means [10]). It is not surprising then that data-logging is now part of the school curriculum for England and Wales at Key Stages 1-3 (ages 5 to 14 years) [1].

Research within the AIED community has explored how we can design adaptive technology that takes learners' context and potential collaborators into account [2, 5]. Much of this work, grounded in a socio-cultural approach to understanding the learning process, has explored the ways in which technology can adapt to scaffold learners' collaborative interaction [e.g. 2]. We have also noted previously that the introduction of tangible interfaces to collaborative interactions can increase the level of social interaction between collaborators beyond that observed with purely desk-top screen based interfaces [4]. Wireless mobile devices should also allow learners to complete activities, thus freeing them up to think about the underlying scientific concepts and processes.

The educational context in which the technology is to be used is an important design parameter, both because of its impact upon device selection and because of its importance to a socio-cultural approach. Previous research in schools has indicated that the impact of technology is heavily dependent upon the specifics of, and extent to which it is embedded in, the educational culture [12]. Adaptive technologies and context of learning research would suggest that a similar socio-cultural underpinning is appropriate for learning situations that combine and match technologies to the learning task and context. However, it is not clear from the emerging research *how* these technologies might be best combined, matched and applied to support teachers and students. There is work that tries to unpack what it is about tangible and hand-held interfaces that makes a difference to a learner's interactions with them and yet progress towards the construction of a satisfactory theoretical framework is slow [6]; understanding the factors with multiple interfaces raises even more challenges. The focus here is not to unpack the process of learning with multiple technologies, but to address some of the important pragmatic questions that need to be explored first such as: What technologies should teachers invest time in? And what benefits do they provide for both students and teachers?

In this paper we report research that explores how multiple mobile devices provided different opportunities for active and hands-on learning, in real-life situations. In addition, we report on ways in which support with using these types of technologies affects the level of collaborative scientific enquiry achieved, as determined by types of explanations provided by students in the different project contexts. This type of investigation is important to the AIED community if we are to develop intelligent ubiquitous systems that can scaffold learners with resources appropriately targeted to both task and context.

3. The projects

The projects described here provide examples of two different groups of learners in two different contexts exploring their understanding of CO air pollution in a local environment. The sensing and data-logging technology used in both projects enabled a combination of automatic and manual data collection when out on location. Each group was given a 'tea tray' [11], an anemometer, a video camera and map of the local area (see Figure 1). The 'tea tray' was made up of a CO monitor; a Global Positioning System (GPS) location sensor; and a Personal Digital Assistant (PDA) that logged both the CO and GPS data from the other two devices. The anemometer was used to manually collect wind speed, whilst the video camera enabled learners to record their own data collection process.

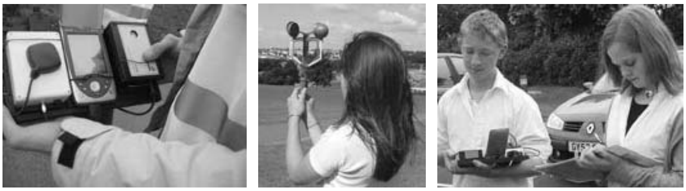


Figure 1 – data-logging technologies

3.1 Project 1 - e-Science

The aim of the e-Science project was to provoke students aged 14-16 years of age to think about how the technologies support their scientific research and learning. Using the domain of CO pollution as an exemplar for this purpose, students learned about factors that might influence local CO levels (e.g. proximity to roads, wind direction and speed, etc.). A guiding principle throughout the sessions was to challenge learners to decide for themselves what e-Science might be. Our intention was for the students' own interest to drive their research and construction of ideas and knowledge.

A total of 42 students worked in small groups of 2, 3 or 4 accompanied by a facilitator (teacher or researcher), and collected their own local CO and wind readings with the 'tea tray' device and anemometer. Students were also asked to make video recordings and were given a map of the campus and locality, around which they could explore. Later in the classroom students reflected on 3D visualisations of the campus overlaid with the CO data they had collected. A total of 12 sessions of 20-30 minutes each took place.

3.2 Project 2 - SENSE

The aim of the SENSE project was to use the exploration of CO pollution to develop scientific enquiry skills among learners aged 13-14 years. Skills included: initial research into a domain; planning an experimental study; articulating hypotheses; hypotheses testing through data-logging; reviewing results and communicating findings to others.

A total of 19 students, working in groups of 3-4, participated in 15 sessions over a 2-week period. Students planned 3 or 4 locations to visit and used identical equipment to the e-Science group (CO 'tea tray', video and wind), with the addition of a paper sheet for logging wind speed and an estimate of its direction. A facilitator (researcher) accompanied each group. In the class-based review sessions, the CO data collected by each group was represented as a graph using a laptop application that synchronised CO readings with video data; students were able to annotate these graphs.

3.3 Students' scientific data collection

By working in groups with a range of devices, the students adopted different roles depending on the device there were using (the 'tea tray', the anemometer, the map or the camera); they were free to swap their device roles if they so chose. The differing goals of each project were reflected in the type of instructions given to students. In general, the groups of students would walk around their survey area, monitoring the continuous read out of CO readings on the PDA. At self-determined intervals they would take a manual wind reading, either stopping to allow their peer to record it, or whilst moving, to check on levels. The CO reading could be automatically noted by pressing a button on the PDA or by writing it on paper, and the wind reading would be written down on a map (e-Science sessions) or wind data collection sheet (SENSE sessions). Maps were annotated by the students to note reading locations as the group moved around.

4. Empirical data analysis

The data collected during the sessions of both projects included video recordings of the data-logging sessions, and logged CO and GPS data. For project 2, SENSE, we also had class based video and annotation data added in the review sessions.

In the analysis of this data, we focus on the following research questions:

- What types of interactions were afforded by the functionality and physical attributes of the different devices?
- What types of group interactions and scientific enquiry activities did students engage in with and around the devices and during subsequent reflective review?

To analyse the videotapes, we produced transcripts and created time-related activity maps (see Figure 2). The activity maps enabled us to build up a picture of the roles played by the different resources, both participants and technological artefacts, in each of the learning situations we investigated. They enable us to unpack indicative ways in which these resources interact and impact upon the nature of the learning activity that occurs; indicative because we are dealing with real world empirical studies rather than carefully controlled lab or classroom based work. However, they are still important for framing the nature of future work, provide guidance for educators wanting to use wireless, mobile and hand-held technology in their teaching and guidance for those involved in the design of such technologies.

Location
Proximity
Group size
Static / Mobile / Class
Role change
Facilitator input
Hypothesis (I / E)
Plan / direct
Comm of readings
Breakdown
Distraction
Discussions

Activity maps provided activity overviews that we used to determine patterns and trends in the behaviour of participants. Creation of the maps required charting the learners' interactions with each other and with the data-logging devices. Interactions were categorised to explore the nature of the scientific activities they took part in and the ideas generated whilst using particular types of technology. A segment of an activity map is shown left, superimposed with the actual CO graph, synchronised, at the bottom.

Aspects noted on the maps included: a breakdown of the type of comments made by each person within the group (including facilitator) and different co-operative and collaborative behaviours, e.g. suggesting where to test for CO or communicating readings to the group.

Figure 2 - Activity map

In the following section we discuss the findings from this analysis and the implications arising from them.

5. Findings

The findings covered in this section focus on three areas relating to the research questions above: the nature of the interactions between students and their devices; factors affecting the way groups co-operated around and with devices; and the nature of the scientific inquiry processes that learners engaged in to procure and explain the data-logged results. We discuss examples that illustrate the combined devices' contribution to a collaborative air pollution exploration within the group, and illustrate examples of collaborative behaviour using the devices, combined with levels of facilitator input, to determine effective and non-effective behaviours.

5.1 Nature of interactions between students and devices

Initial analysis of how learners interacted with each device focused on the level of the individual learner's contribution. The patterns of interaction that emerged across all sessions indicated that each device's function and physical attributes afforded a different way of interacting with it.

5.1.1 The importance of the level of control: "Let's note the high readings"

The 'tea tray' PDA automatically logged and displayed CO readings, whilst the anemometer readings were taken less frequently and not always by the person holding it. In contrast with the 'tea tray', students frequently played with the anemometer, blowing at it to get a high wind reading or trying to get the 'spoons' to rotate as fast as possible. Selective sampling also occurred, whereby the highest reading was recorded each time, as the students believed this was the most impressive figure to note. For example, the anemometer holder, on a very windy day, was heard saying: *"It was 6 [metres per second] a minute ago"*, encouraging the noting of that figure rather than the current 0 or 1 reading.

This presents an interesting set of trade-offs for design. Whilst the ability to control and explore a device is important to understand the properties of what is being measured and learn about accurate science data-logging, an automated wind-measuring device would reduce the level of control the wind person has over readings, and would give a more accurate value at the time requested. A digital device would have the added benefit of being more easily synchronised with the CO data for classroom reflection.

5.1.2 The value of a data history: Carbon Monoxide data-logger

The person holding the 'tea tray' played a key role since the user-interface of the 'tea tray' was only visible to the person holding it and, therefore, the group had to rely on that person to communicate the CO values. Across all sessions then, engagement with the device was high and the person allocated to this device tended to keep the group informed of any changes in CO levels. However, we also saw communication breakdowns occur when there was no change in CO levels to report, and when the person carrying the 'tea tray' was too shy to take the initiative and call out a reading without being prompted.

While the calling out of CO values depended partly on the personality of the student holding the 'tea tray', the addition of a trend graph was found to be particularly useful when the CO person had been quiet or distracted. For example, the wind device holder took on the role of reporting CO in the absence of the CO person or video person doing this:

> Wind device holder comes over to look at CO: *"how come it's gone up so much?"*
> Camera person: *"it went up to about 6.5 [parts per million]... yeah that engine..."*
> Wind person [gets Camera person to move camera on to him]: "*The Carbon Monoxide went up greatly because there was a parked van with its engine running still by us*".

The trend graph in this instance enabled the wind device holder to determine how quickly CO had risen, giving him a timescale for reasoning out why the rise occurred.

5.1.3 The potential for distraction: Video camera

We found the camera person tended to be the least informed about the data readings as they stood back to capture the group. They were often heard requesting readings from the two data-logging device people, and asking *"what are we doing now?"*. We also noted a strong tendency for the camera person (more than the other roles) to be distracted away from their task of filming by other peers, workmen, teachers and members of the public. To reduce this dis-engagement with the task we would suggest the video person is encouraged to take on an 'interviewer' type role. This could reduce the physical space between group and camera person, give more purpose for all members of the group to narrate their activity to camera, and reduce the likelihood of distancing any individual from the group.

5.2 Collaborating and engaging in scientific enquiry

5.2.1 The importance of facilitation

The facilitator role was important in shaping group interactions during the data collection sessions by engaging the group and encouraging critical thinking. The differences in focus of the two projects resulted in different facilitator emphases, for example allowing free exploration (e-Science session) as opposed to the testing of CO at pre-planned locations, interspersed with on-the-fly stops (SENSE session). In particular, effective actions were identified as prompting for CO and wind readings; for hypotheses to explain CO readings; for locations where CO levels would be high; and encouraging students to contrast with previous places visited.

In response to a SENSE facilitator asking why they thought the busy road had not produced as much change as predicted, the students engaged in 4 minutes of discussion, resulting in three hypotheses being verbalised on the effect of cars; buses; and diesel versus petrol engines on CO. These developed hypotheses were not the focus of the e-Science sessions and did not occur in those sessions. Poor facilitation occurred on both projects when the facilitator's intervention was minimal, resulting in students data-logging without questioning their readings, nor developing explanations or interacting with each other beyond carrying out minimal task activities.

5.2.2 The role of in-situ explanations and reflective review

The effect of environmental context on explanations was salient in both groups. Once the students had started to collect readings, they gave a range of explanatory reasons including reference to the presence (or absence) of wind speed and direction, car traffic, larger vehicles, and proximity to vehicle exhausts. Some groups were further motivated to control conditions to test out their developing ideas: one SENSE group used a pedestrian crossing to stop traffic and see whether a build up of CO occurred. This led them on to consider the direction of wind movement to determine whether they had chosen the best location relative to the queuing traffic, and then reposition the 'tea tray' down-wind. Julie summarised her thoughts: "*at the traffic lights cars stop then they start again so they must go, chuck a lot more carbon monoxide out.*"

The technology used by both projects described here enabled students to combine readings from different devices, to pool the ideas they had formed from their different device perspectives, to re-formulate hypotheses and to adjust their data collection plans in order to test these hypotheses. When they did return to the classroom they could reflect upon their experiences, review their findings and their data collection skills. SENSE students reflected upon and learned how to improve the process of collecting data by reviewing their video and data. When making annotations students often referred to their lack of good filming skills, and occasionally found, for example, that a high CO reading had occurred and gone unnoticed. It was instances such as these that encouraged them to revisit parts of the video recordings to identify exactly what was happening. In this way the review session helped students analyse the data in a more productive way than the visual graphing of data points alone [1].

A key value that arose from the review sessions was that groups developed their hypotheses and adjusted predictions for CO levels in preparation for the second data collection session. In the second data-logging session as compared to the first, most students engaged in more narration activities, with spontaneous sharing of readings within the group, more data requests of each other and fewer incidences of distraction. For example one group's narration and direction comments increased by 150%, and their communication of readings increased by 200%.

6. Discussion, conclusions and future work

We have presented illustrative findings from two related projects, which identify the factors affecting group interactions around hand-held devices from the perspective of single and multi-session investigations by students. We found that the major impacts on device activity were: the ability to control and explore devices, the availability of trend data and the amount of distraction created by device roles. The type of facilitator input affected group co-operation; and the combination of in-situ data collection sessions interspersed with reflective review produced valuable opportunities to develop group ideas and hypotheses. From our findings we have gained an increased understanding of what needs to be done to facilitate learning around such technologies. We would recommend the following considerations in designing similar data-logging experiences, these include pointers for the development of software enabled scaffolding interventions:

- Consider the trade-offs between a controllable interface versus an accurate data log.
- Provide trend data particularly for variable data such as wind readings.

- Remind learners to vocalise information regularly with peers, which could be given through PDA-initiated prompts to answer related questions.
- Consider the use of larger screens and audio displays to allow all group members to be aware of general trends in data-logged readings.
- Scaffold appropriate facilitator input e.g. via PDA using a suggested question for group discussion, triggered by location, incorporating current data-logged values.

Our experience clearly shows the need for future work to focus on the effects of building 'roles' around devices and of facilitator input. For example, what kind of guidance should facilitators provide, and how much? Could some of this input be mediated by a combination of user modelling, combined with location sensing, and hypothesis knowledge – and should it go directly to the students, or prompt the facilitator to ask students? One aim would be to build relationships within the group over time to create a more talkative, thinking, creative dialogue to enhance learning and collaboration by each group member.

7. Acknowledgements

We would like to thank Danae Stanton Fraser, Sara Price, Ella Tallyn, the SENSE project collaborators at Nottingham, session participants including David Daniels, Steve Rogers, teachers and pupils at Varndean and Hove Park Schools, Portslade Community College. The e-Science project was part of the Equator IRC (www.equator.ac.uk). The SENSE project was funded by the JISC.

8. References

[1] DfES (2005). The Standards Site. Department for Education and Skills, UK Government, http://www.standards.dfes.gov.uk verified 9 February 2005

[2] Greer, J., McCalla, G., Cooke, J., Collins, J., Kumar, V., Bishop, A. and Vassileva, J. (1998) The Intelligent Helpdesk: Supporting Peer-Help in a University Course. In *Proceedings of 4th International Conference on Intelligent Tutoring Systems,* 494-503

[3] Luchini, K., Quintana, C., Curtis, M., Murphy, R., Krajcik, J., Soloway, E. and Suthers, D. (2002). Using Handhelds to Support Collaborative Learning. *Computer Support for Collaborative Learning,* 704-705

[4] Luckin, R., Connolly, D., Plowman, L. and Airey, S. (2003). With a little help from my friends: Children's interactions with interactive toy technology. *Journal of Computer Assisted Learning (Special issue on Children and Technology),* 165-176

[5] Murray, T. and Arroyo, I. (2002) Towards Measuring and Maintaining the Zone of Proximal Development in Adaptive Instructional Systems. In *Proceedings of Sixth International Conference on Intelligent Tutoring Systems,* Springer

[6] Price, S., Rogers, Y., Scaife, M., Stanton, D. and Neale, H. (2003). Using 'tangibles' to promote novel forms of playful learning. *Interacting with Computers,* 15(2), 169-185

[7] Rogers, Y., Price, S., Randell, C., Stanton Fraser, D., Weal, M. and Fitzpatrick, G. (2005). Ubi-learning Integrates Indoor and Outdoor Experiences. *Communications of the ACM,* 48(1), 55-59

[8] Roschelle, J. and Pea, R. (2002) A walk on the WILD side: How wireless handhelds may change CSCL. In *Proceedings of Computer-Support for Collaborative Learning,* 51-60

[9] Sims Parr, C., Jones, T. and Songer, N. (2002) CyberTracker in BioKIDS: Customization of a PDA-based Scientific Data Collection Application for Inquiry Learning. In *Proceedings of Keeping Learning Complex: The Proceedings of the Fifth International Conference of Learning Sciences (ICLS),* Erlbaum, 574-581

[10] Stanton Fraser, D., Smith, H., Tallyn, E., Kirk, D., Benford, S., Rowland, D., Paxton, M., Price, S. and Fitzpatrick, G. (in press) The SENSE project: a context-inclusive approach to studying environmental science within and across schools. Accepted for publication for *CSCL'05*

[11] Steed, A., Spinello, S., Croxford, B. and Greenhalgh, C. (2004) e-Science in the Streets: Urban Pollution Monitoring. In *Proceedings of 2nd UK e-Science All Hands Meeting 2003*

[12] Wood, D., Underwood, J. and Avis, P. (1999). Integrated Learning Systems in the Classroom. *Computers and Education,* 33(2/3), 91-108

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Implementing a Layered Analytic Approach For Real-Time Modeling of Students' Scientific Understanding

Ron STEVENS, Amy SOLLER
IMMEX Project, UCLA, 5601 W. Slauson Ave, #255, Culver City, CA. 90230
immex_ron@hotmail.com, asoller@ida.org

Abstract. We have developed layered analytic models of how high school and university students construct, modify and retain problem solving strategies as they learn to solve science problems online. First, item response theory modeling is used to provide continually refined estimates of problem solving ability as students solve a series of simulations. In parallel, the strategies students apply are modeled by self-organizing artificial neural network analysis, using the actions that students take during problem solving as the classifying inputs. This results in strategy maps detailing the qualitative and quantitative differences among problem solving approaches. Learning trajectories across sequences of student performances are developed by applying Hidden Markov Modeling to stochastically model problem solving progress through the strategic stages in the learning process.

Using this layered analytical approach we have found that students quickly adopt preferential problem solving strategies, and continue to use them up to four months later. Furthermore, the approach has shown that students working in groups solve a higher percentage of the problems, stabilize their strategic approaches quicker, and use a more limited repertoire of strategies than students working alone. In this paper, we also describe our ongoing and future work in developing an online collaborative learning environment that both models the group interaction and identifies which individual student contributions might contribute to increased achievement.

Keywords. Chemistry, Artificial Neural Networks, Hidden Markov Modeling, Student Modeling.

Introduction

Dynamically modeling how students approach and solve scientific problems at various levels of detail and at different points in time could provide evidence of a student's changing understanding of a task, as well as the relative contributions of different cognitive processes to the student's problem solving strategy [1] [2]. Given sufficient detail, such models could extend our understanding of how gender, prior achievement, classroom practices, and other student characteristics differentially influence performance and participation in complex problem-solving environments [3]. If the models had predictive properties, they could also provide a framework for directing feedback to improve learning through direct teacher support, collaborative learning interventions [4], or even appropriately trained pedagogical agents [5].

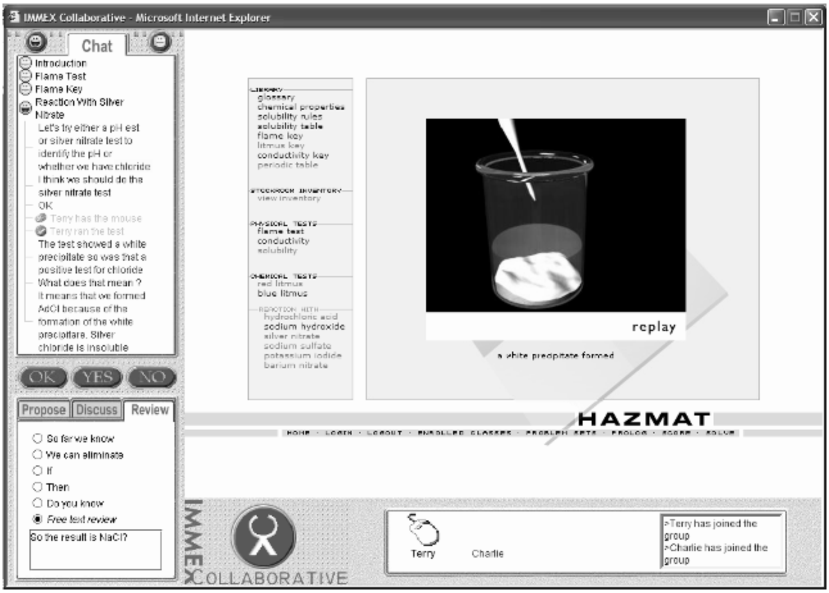
The idea of 'learning trajectories' is a useful context for thinking about the development of such models [6]. These trajectories are based on the different ways that novices and experts think and perform in a domain, and can be viewed as defining stages of understanding as students develop experience [7]. Not all novices solve problems in the same way, nor do they follow the same path at the same pace as they develop an understanding of the domain. In this research, we apply a combination of machine learning methods to identify the variety of strategies that novices use in developing competence, and link these strategies to the stages they go through as they learn. We describe how we have coupled an online learning environment with a layered system of analytic tools to dynamically model the following measures:

- What is the strategic sophistication of a student at a particular point in time (a performance measure)?
- How did the student arrive at this level (a progress measure)?
- How will s/he likely progress with more practice/experience (a predictive measure)?
- How long will the students retain this strategic level (a retention measure)?
- What learning/instructional interventions will most effectively accelerate each student's learning (interventions)?

The next section introduces the problem solving environment, and addresses the first two points regarding performance and progress. Section 2 then discusses how our combination of probabilistic approaches can be used to predict future student performance and content retention. In section 3, we describe what we have learned about students' shifting dynamics in strategic reasoning, and describe our future work in applying collaborative learning methods to encourage students to adopt effective problem solving strategies.

1. Tasks, Approaches and Populations

In this paper, we describe how the IMMEX (Interactive Multi-Media Exercises) problem solving environment has facilitated our study of student strategy adoption and persistence [8] [9]. IMMEX problem solving follows the hypothetical-deductive learning model of scientific inquiry [10] in which students frame a problem from a descriptive scenario, judge what information is relevant, plan a search strategy, gather information, and eventually reach a decision that demonstrates understanding. We have chosen the IMMEX problem set termed Hazmat to model strategic development because it challenges students in conducting qualitative chemical analyses and provides evidence of their ability (Figure 1). The problem begins with a multimedia presentation, explaining that an earthquake caused a chemical spill in the stockroom; the student's challenge is to identify the chemical. The problem space contains 22 menu items for accessing a Library of terms, the Stockroom Inventory, or performing Physical or Chemical Testing. When the student selects a menu item, she verifies the test requested and is then shown a presentation of the test results (e.g. a precipitate forms in the liquid) When students feel they have gathered adequate information to identify the unknown they can attempt to solve the problem. To ensure that students gain adequate experience, this problem set contains 34 cases that can be performed in class, assigned as homework, or used for testing.



Let's try either a pH est or silver nitrate test to identify the pH or whether we have chloride

I think we should do the silver nitrate test

OK

Terry has the mouse

Terry ran the test

The test showed a white precipitate so was that a positive test for chloride

What does that mean ?

It means that we formed AdCl because of the formation of the white precipitare. Silver chloride is insoluble

○ So far we know
○ We can eliminate
○ If
○ Then
○ Do you know
● Free text review

So the result is NaCl?

a white precipitate formed

>Terry has joined the group
>Charlie has joined the group

Fig. 1. ***HAZMAT*** This screen shot of *Hazmat* shows the menu items down the left side of the main "Hazmat" window on the screen and a sample test result (the result of a precipitation reaction). In this figure, the IMMEX problem set has been embedded within a collaborative learning environment, allowing groups of students to chat using sentence openers (left-hand panel of the screen) and share mouse control (bottom panel, see section 3).

By having students perform multiple cases that vary in difficulty, student ability can be obtained by Item Response Theory (IRT), an analysis technique which relates characteristics of items (item parameters) and individuals (latent traits) to the probability of a positive response (such as solving a case). Using IRT, pooled data about whether or not a student solved a particular case on the first attempt (rating = 2), on the second attempt (rating = 1), or failed to solve the case (rating = 0) is first used to calibrate all of the items, and then and to obtain a proficiency estimate for each student [11]. As shown in Figure 2, the cases in the problem set are of a range of difficulties, and include a variety of acids, bases, and compounds that give either a positive or negative result when flame tested. The distribution of student proficiency measures shows that the problems cover an appropriate range of difficulties providing accurate estimates of student ability.

1.1 Identifying and Modeling Strategic Approaches

Although IRT is useful for ranking the students by the effectiveness of their problem solving, it does not provide a strategic measure of student problem solving. For this, we apply Artificial Neural Network (ANN) analysis procedures. As students navigate the problem spaces, the IMMEX database collects timestamps of each student selection. The most common student approaches (i.e. strategies) for solving Hazmat are identified with competitive, self-organizing artificial neural networks [12] [13] [9] [14]. These ANNs input the students' selections of menu items as they solve the problem, and output a topological ordering of the neural network nodes according to the structure of the data. The geometric distance between nodes then becomes a metaphor for strategic similarity. Often we use a 36-node neural network, in which each node is visualized by a histogram (Figure 3 A). The histograms show the frequency of items selected for the student performances classified at that node. Strategies are defined by the items that are always selected for performances at that node (i.e. with a frequency of 1) as well as items ordered more variably.

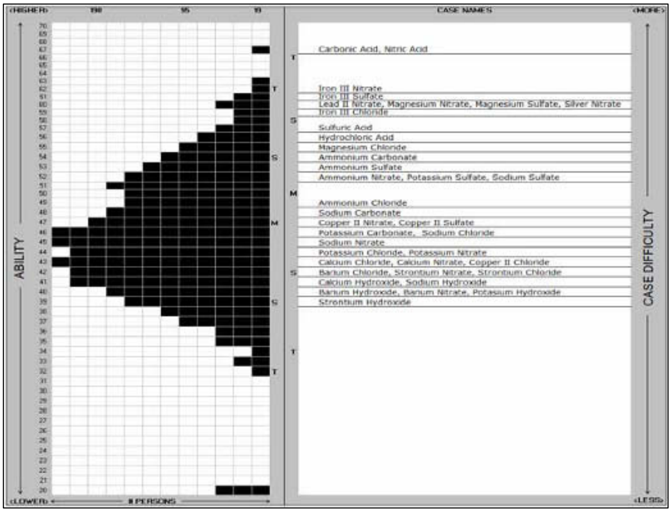


Figure 2. Levels of Problem Difficulty. The case item difficulties were determined by IRT analysis of 28,878 student performances. The problem difficulty begins with the easiest cases at the bottom and increases towards the top. The distribution of student abilities is shown on the left. The highest ability students reside at the top and ability decreases towards the bottom. For each graph, **M** indicates the mean, **S**, the standard deviation, and **T** two standard deviations.

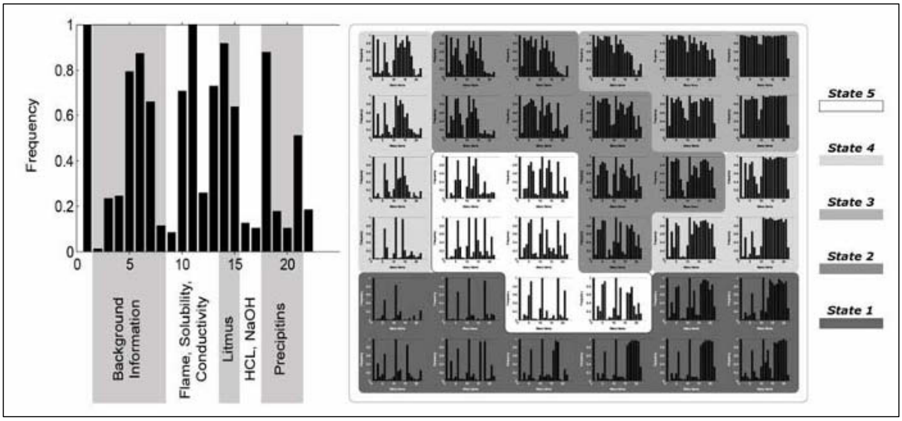


Fig. 3. Sample Neural Network Nodal Analysis. A. This analysis plots the selection frequency of each item for the performances at a particular node (here, node 15). General categories of these tests are identified by the associated labels. This representation is useful for determining the characteristics of the performances at a particular node, and the relation of these performances to those of neighboring neurons. **B.** This figure shows the item selection frequencies for all 36 nodes following training with 5284 student performances.

Figure 3 B is a composite ANN nodal map of the topology of performances generated during the self-organizing training process. Each of the 36 graphs in the matrix represents one node in the ANN, where similar student problem solving performances become automatically clustered together by the ANN procedure. As the neural network is trained with vectors representing the items students selected, it is not surprising that a topology develops based on the quantity of items. For instance, the upper right hand of the map (nodes 6, 12) represents strategies where a large number of tests have been ordered, whereas the lower left corner contains strategies where few tests have been ordered.

Once ANN's are trained and the strategies represented by each node are defined, new performances can be tested on the trained neural network, and the node (strategy) that best matches the new performance can be identified and reported. The strategies can be aggregated by class, grade level, school, or gender, and related to other achievement and demographic measures.

1.2 Hidden Markov Model Analysis of Student Progress

Artificial neural network analyses provide point-in-time snapshots of students' problem solving; however, any particular strategy may have a different meaning at a different point in a learning trajectory. More complete models of student learning should therefore take into account the changes in student strategies with practice over time.

To model student learning progress over multiple problem solving episodes, students perform multiple cases in the 34-case Hazmat problem set, and we classify each performance independently with the trained ANN. Some sequences of performances localize to a limited portion of the ANN topology map. For instance the nodal sequence {32, 33, 28, 33, 33} suggests only small shifts in strategy with each new performance. In this research we use Hidden Markov Modeling (HMM) to extend our preliminary results to more predicatively model student learning pathways.

Markov models are used to model processes that move stochastically through a series of predefined states over time [15] [16] [17] [18]. In applying this process to modeling student performance, we postulate that a number of unknown states exist in the dataset representing strategic transitions that students may pass through as they perform a series of IMMEX cases. These states might represent learning strategies that task analyses suggest students may pass through while developing competence [19]. For most IMMEX problem sets, a postulated number of states between 3 and 5 have produced informative models. Then, similar to the previously described ANN analysis, exemplars of sequences of strategies (ANN node classifications) are repeatedly presented to the HMM modeling software to develop temporal progress models. The resulting models are defined by a transition matrix that shows the probability of transiting from one state to another, an emission matrix that relates each state back to the ANN nodes that best represent student performances in that state, and a prior matrix which estimates the most likely (starting) states within which students might begin their learning and thought processes.

The mapping between ANN nodes and HMM states is shown in Figure 4. The nodes associated with each state are overlaid and highlighted on the 6 x 6 neural network grid. The 5 different HMM states reflect different strategic approaches with different solution frequencies, meaning that students who adopt strategies in some states tend to perform better than other students. For example, state 1 is an absorbing state that represents a limited strategy in which students use Background information minimally, and the different Test Items variably. This qualitative assessment is done by analyzing the group of ANN nodes that map to the states that students transition through as they are learning. State 2 shows balanced usage of Background Information and Test Items, but little use of precipitation reactions. State 3 shows a very prolific approach in which students use all the menu items extensively. State 4, like State 2, is a transitional state, but with more focused testing. Transitional states are those that students are likely to transition out of while they are learning. State 5 has the highest solution frequency, which makes sense because its ANN histogram profile suggests that students in this state pick and choose certain tests, focusing their selections on those tests that will help them obtain the solution most efficiently.

HMM models of student strategy progression also enable us to make predictions regarding the student's learning trajectory. We developed a procedure that compares the 'true' state values of a student's subsequent performance with the next state predicted by the HMM. This procedure produces an accuracy of 50% early in a sequence of performances, and increases to 75-90% as more cases are attempted.

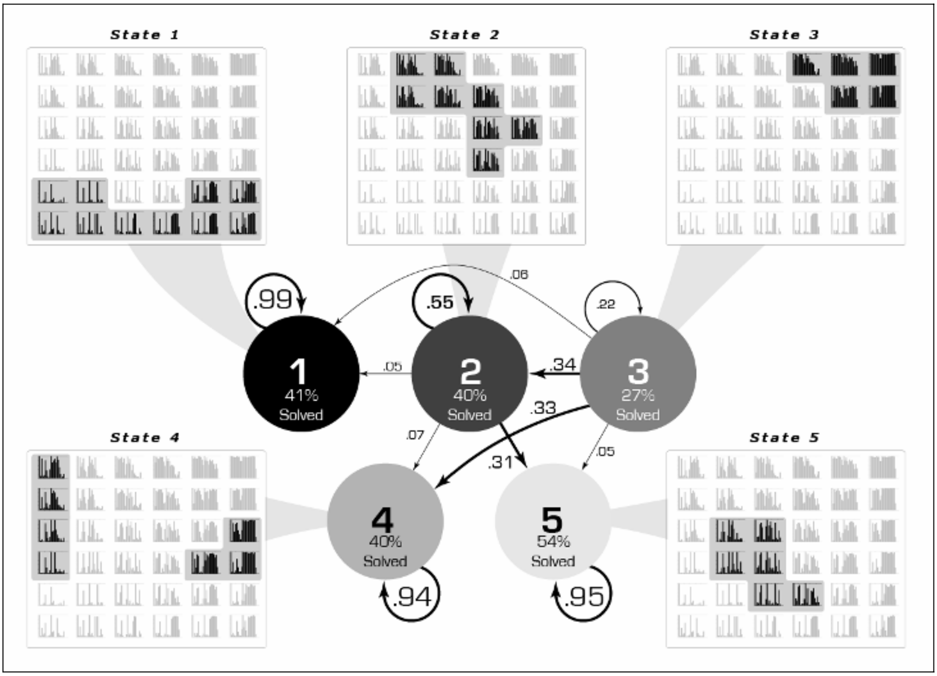


Figure 4. HMM Transition and Emission Matrices. This figure illustrates the transition and emission matrices obtained by training the HMM with 1790 student performances and shows the likelihood that students will transit from one state to another. Looking along the curved lines, States 1, 4, and 5 appear stable, suggesting that once students adopt these strategies, they are likely to continue to use them. In contrast, students adopting State 2 and 3 strategies are less likely to persist with those approaches, and more likely to adopt other strategies. The highlighted graphs in each map indicate which ANN nodes are most frequently associated with each state. The solution frequencies represent the percentage of students who obtained the correct answer on their first attempts.

2. Results

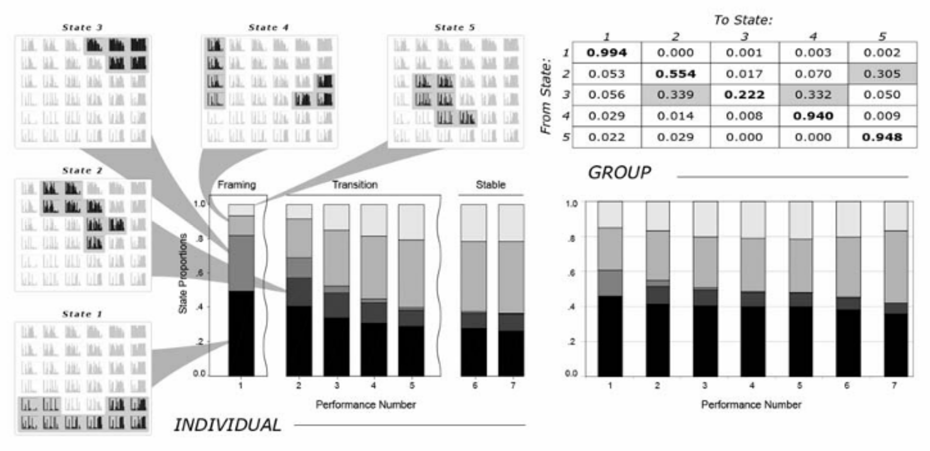
2.1 Dynamics of State Changes With Experience

When students perform a series of cases, their strategic approaches shift over the first 3 to 4 performances, and then stabilize as they develop strategies with which they are comfortable. An example of these dynamics is shown in Figure 5: as 7196 students performed 7 Hazmat cases, the number of students using strategies characterized by States 1 and 3 decreased over time, while State 2 strategies increased initially and then decreased gradually over the last 6 performances, and States 4 and 5 generally increased. Across these 7 Hazmat performances, the overall solution frequency (for first and second attempts) increased from 53% (case 1) to 62% (case 7), suggesting that most students generally improve over time while solving Hazmat cases.

The solution frequencies at each state provide an interesting view of student progress. For instance, if we compare the earlier differences in solution frequencies with the most likely state transitions from the matrix shown in Figure 4, we see that most of the students who enter State 3, having the lowest problem solving rate (27%), will transit either to State 2 or 4, and increase their solution frequency by 13% on average. Students performing in State 2 are more likely than those in State 4 to transit to State 5 (with a 14% increase in solution frequency). From an instructional point of view, these results suggest that we might guide students who are performing in State 3 toward State 2 rather than State 4 strategies. The effect of a successful instructional intervention in this case would give the student a 30% greater chance to attain a State 5 strategy, which would increase their problem solving effectiveness by 27%.

2.2 Effects of Collaborative Grouping on Learning Trajectories

In this section, we discuss our preliminary results in using the same ANN and HMM methods described in the first part of this paper to model collaborative learning groups. Consistent with the literature on collaborative learning [20], we found that having students work in collaborative groups significantly increased their solution frequency from 39% to 49%. This result is also reflected by the groups' strategic learning trajectories. Figure 5 A, (discussed in detail in the previous section), illustrates the state dynamics for students working individually, and is characterized by the extensive use of State 3 strategies early in the problem solving process, with transitions through State 2, and stabilization on States 1, 4, and 5. Most individuals stabilize their strategy usage by the 5th performance.



From State:	To State: 1	2	3	4	5
1	**0.994**	0.000	0.001	0.003	0.002
2	0.053	**0.554**	0.017	0.070	0.305
3	0.056	0.339	**0.222**	0.332	0.050
4	0.029	0.014	0.008	**0.940**	0.009
5	0.022	0.029	0.000	0.000	**0.948**

Figure 5, Learning trajectories for individuals (A) and groups (B). The bar chart tracks the changes in all student strategy states (n=7196) across seven *Hazmat* performances. Mini-frames of the strategies in each state are shown for reference.

In contrast, collaborative learners use more State 1 strategies (limited and / or guessing approaches), and stabilize their strategy usage by the 3rd performance (Figure 5 B). A comparison of the state distributions shows that groups use fewer of the transitional states 2 and 3, and instead progress rapidly to the more stable states. The percentage of cases that students solve in States 1 and 4 increases significantly when they learn collaboratively (from 40-52% and 39 to 45% respectively), whereas remained the same for State 5 (54-57%).

3. Discussion

These studies were motivated by our interest in understanding students' shifting dynamics in strategic reasoning as they gain problem solving experience: an understanding that could perhaps be extended to develop targeted feedback to teachers and students to improve learning. Our analytic approach is necessarily multilayered to address the broad needs set out in the introduction to this paper. The analytic model combines three algorithms (IRT, ANN and HMM) that, when integrated with problem set design and classroom implementation decisions, provides an extensible system for modeling strategies and formulating interventions. When combined, these algorithms provide a considerable amount of real-time strategic performance data about the student's understanding, including the IRT person ability estimate, the current and prior strategies used by the student in solving the problem, and the strategy the student will most likely use next, all of which provide information important to constructing detailed models of scientific understanding development.

Most students approach the first *Hazmat* case by selecting either an extensive (State 3), or limited (State 1) amount of information. The State 3 approaches would be appropriate for novices on the first case as they strive to define the boundaries of the problem space. Persisting with these strategies, however, would indicate a lack of understanding and progress. State 1, in which students select a limited amount of information, has only a mediocre solution rate, and is an absorbing state; once adopted students are unlikely to change from it.

As students gain experience, their strategies change. Background information that was needed earlier may no longer be needed, and students begin to develop their own favorite approaches based on knowledge, experience, motivation, and prior experiences. The main transition states are States 2 and 3. When students transition out of State 3, this suggests that they are learning: the transition matrix shows that these students are likely to switch to States 2 or 4, thus increasing their likelihood of solving the case from 27% to 40%. The main difference between States 2 and 4 is that State 2 approaches include access to both test and background information, whereas State 4 approaches are primarily data driven.

The states on which students stabilize reflect the level of competence and the type of strategic approach that they eventually feel comfortable using in a particular situation. These approaches are the ones that would most often be recognized by teachers. For individuals, stabilization occurs with States 1, 4 and 5 strategies. State 4 is interesting in several regards. It differs from the other states in that the strategies it represents are located at distant points on the ANN topology map, whereas the nodes comprising the other states are contiguous. The State 4 strategies represented by the left hand of the topology map are very appropriate for the set of cases in Hazmat that involve flame test positive compounds, whereas those strategies on the right side are more appropriate for flame test negative compounds (where more extensive testing for both the anion and cation are required). This suggests that students using State 4 strategic approaches may have mentally partitioned the Hazmat problem space into two groups of strategies, depending on whether the initial flame test is positive.

Students working collaboratively improve their problem solving (by IRT) and stabilize their strategies faster than students working alone, begging the usual question about why collaborative learning in this case is effective. Some indication comes from the different state distributions describing individual and group performances. Group performances mainly stabilize with State 1, which appears to be strategically heterogeneous in that it contains student performances representing guessing (with a low solved rate), as well as very limited, but effective strategies. Collaborative learners performing in this state can be successful problem solvers, and tend not to need states 2 and 3, suggesting that collaboration with peers encourages students to make the appropriate transitions within states 1 and 2, rather than explicitly transiting through them.

An important next step will be analyzing the qualitative and quantitative group interaction data to understand how the collaboration affects these learning trajectory changes. We are beginning to develop such web-based collaboration models by integrating IMMEX into a web-based scientific inquiry environment (see Figure 1). Collaborative IMMEX allows groups of students to communicate through a chat interface (with specially designed sentence openers), and share workspace control while solving Hazmat and other IMMEX problems [21]. By monitoring and assessing the collaborative interaction [18], and comparing it to the problem solving outcomes defined by the HMM strategic models, we hope to not only determine more precisely what aspects of the collaboration modulate problem solving strategies, but also strategically pair individuals in combinations that our models suggest will enhance the learning of both partners.

Supported in part by grants from the National Science Foundation (ROLE 0231995, DUE 0126050, HRD-0429156), the PT^3 Program of the U.S. Department of Education (P342A-990532), and the Howard Hughes Medical Institute Precollege Initiative.

References

1. Anderson, J.D,(1980). Cognitive Psychology and its Implications. San Francisco: W.H. Freeman.
2. Mayer, R.E., (1998). Cognitive, metacognitive and motivational aspects of problem solving. Instructional Science 26: 49-63.
3. Fennema, E. Carpenter, T., Jacobs, V., Franke, M., and Levi, L. (1998). Gender differences in mathematical thinking. Educational Researcher, 27, 6-11.
4. Case E., 2004. The Effects of Collaborative Grouping on Student Problem Solving in First Year Chemistry. Unpublished thesis.
5. Arroyo, I., Beal, C., Murray, T., Walles, A., ad Woolf, B. (2004). Web-Based Intelligent Multimedia Tutoring for High Stakes Achievement Testing. LNCS, 3220, 468-477 2004
6. Lajoie, S.P. (2003). Transitions and trajectories for studies of expertise. Educational Researcher, 32: 21-25.
7. VanLehn, K., (1996). Cognitive skill acquisition. Annu. Rev. Psychol 47: 513-539
8. Underdahl, J., Palacio-Cayetano, J., & Stevens, R., (2001). Practice makes perfect: Assessing and enhancing knowledge and problem-solving skills with IMMEX Software. Learning and Leading with Technology. 28: 26-31
9. Stevens, R., Wang, P., & Lopo, A. (1996). Artificial Neural Networks can distinguish novice and expert strategies during complex problem solving. JAMIA vol. 3 Number 2 p 131-138
10. Lawson, A.E. (1995). Science Teaching and the Development of Thinking. Wadsworth Publishing Company, Belmont, California
11. Linacre, J.M. (2004). WINSTEPS Rasch measurement computer program. Chicago. [http://www.winsteps.com]
12. Kohonen, T., (2001). Self Organizing Maps. 3rd extended edit. Springer, Berlin, Heidelberg, New York
13. Stevens, R.H., and Najafi K. (1993). Artificial Neural Networks as adjuncts for assessing medical students' problem-solving performances on computer-based simulations. Computers and Biomedical Research 26(2), 172-187

14. Stevens, R.H., Ikeda, J., & Casillas, A., Palacio-Cayetano, J., and S. Clyman (1999). Artificial Neural Network-based performance assessments. Computers in Human Behavior, 15: 295-314

15. Rabiner, L., (1989). A tutorial on hidden Markov Models and selected applications in speech recognition. Proc. IEEE, 77: 257-286

16. Murphy, K. [http://www.ai.mit.edu/~murphyk/Software/HMM/hmm.html].

17. Soller, A., & Lesgold, A. (2003). A Computational Approach to Analyzing Online Knowledge Sharing Interaction. Proceedings of Artificial Intelligence in Education, 2003. Australia, 253-260

18. Soller, A. (2004). Understanding Knowledge Sharing Breakdowns: A Meeting of the Quantitative and Qualitative Minds. Journal of Computer Assisted Learning, 20, 212-223.

19. Stevens, R.H., Soller, A., & Johnson, D. (2005). Predictions and probabilities: Modeling the development of scientific problem solving skills. Cell Biology Education, vol. 4 Number 1. [http://www.cellbioed.org]

20. Brown, B., & Palincscar, A., (1989). Guided, cooperative learning and individual knowledge acquisition. In L. Resnick (Ed.), *Knowing, learning and instruction: Essays in honor of Robert Glaser.* Hillsdale, NJ: Lawrence Erlbaum Associates.

21. Giordani, A., & Soller, A. (2004). Strategic collaboration support in a web-based scientific inquiry environment. European Conference on Artificial Intelligence, “Workshop on Artificial Intelligence in Computer Supported Collaborative Learning”, Valencia, Spain.

22. Webb, N., & Palincsar, A. (1996). Group processes in the classroom. In D. Berlmer & R. Calfee (Eds.), *Handbook of Educational Psychology* (pp. 841-873). New York: Simon & Schuster Macmillan.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Long-Term Human-Robot Interaction: The Personal Exploration Rover and Museum Docents[1]

Kristen Stubbs [a,2], Debra Bernstein [b], Kevin Crowley [b], and Illah Nourbakhsh [a]
[a] *Robotics Institute, Carnegie Mellon University*
[b] *Learning Research and Development Center, University of Pittsburgh*

Abstract. As an increasing number of robots have been designed to interact with people on a regular basis, research into human-robot interaction has become more widespread. At the same time, little work has been done on the problem of long-term human-robot interaction, in which a human uses a robot for a period of weeks or months. As people spend more time with a robot, it is expected that how they make sense of the robot - their "cognitive model" of it - may change over time. In order to identify factors that will be critical to the future development of a quantitative cognitive model of long-term human-robot interaction, a study was conducted involving the Personal Exploration Rover (PER) museum exhibit and the museum employees responsible for it. Results of the study suggest that these critical factors include how people experience successes and failures with the robot (as opposed to how they understand its capabilities) and how people anthropomorphize the robot and talk about anthropomorphization.

Keywords. human-robot interaction, informal learning, educational robotics

1. Introduction

The number of robots designed to interact with humans has increased in recent years, giving rise to the field of "human-robot interaction" as a domain of scientific interest [1]. Within this domain, researchers have designed robots to interact and collaborate with humans in a variety of ways. For example, the Sony *AIBO* is intended for use as a toy [2], *Robovie* was designed to help teach English to Japanese schoolchildren [3], and still other robots have been created to assist humans with urban search and rescue [4].

Despite covering a wide range of activities, it is important to note that most of these robots do not interact with their human users for more than a few minutes or hours at a time. However, if robots are being built with the intention of interacting with people over the long-term, it is crucial to investigate how people understand, model, and interact with robots over long periods of time. This is an interesting and challenging research problem

[1] This work is partially supported by NASA/Ames Autonomy, Intel Corporation, and an NSF Graduate Student Fellowship to K. Stubbs.

[2] Correspondence to: Kristen Stubbs, Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213; Tel.: +1 412 268 8813; E-mail: kstubbs@cmu.edu.

as it requires access to robots that will function properly with minimal maintenance for months on end and at the same time have a rich interaction modality with human beings.

2. Research Goals

The primary goal of this research is to help establish how people's understanding of a robot – their cognitive model of the robot – changes over time. This work can then be used to help generate a quantitative model of long-term human-robot interaction. In order to identify the factors that will be most important in the development of such a model, this study focuses on the human user and how he or she makes sense of a robot after a period of regular interaction lasting weeks or months.

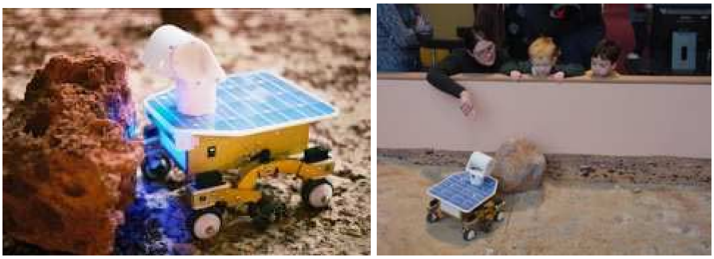
While numerous robots have been designed to be used by humans over long periods of time, few long-term human-robot interaction studies have been conducted at this time. A number of robots have been created that might eventually be used by humans for long periods of time to provide therapy or other assistive services for humans (see [5], [6], [7]); however, none of these robots have been tested with people for long periods of time. One robot that has been studied over a relatively long period of time is *Cero* ([8]). In this study, a motion-impaired user utilized *Cero* to help her carry out various tasks over a period of months; however, this research mainly focused on communication and mediated interaction. The authors' study of the Personal Exploration Rover and museum docents is unique in that it focuses on the relationship between a group of people and a particular type of robot over a period of months, placing emphasis on understanding how the docents' understanding of and interacting with the robot may change over time.

Constructing a complete cognitive model of long-term human-robot interaction is beyond the scope of this study. Instead, the focus of this research is on identifying factors that will play a crucial role in the future development of such models. In order to meet this goal, the authors chose to study the Personal Exploration Rover (PER), a small robot designed to operate in science centers across the United States [10]. The PER was an excellent focus for a long-term human-robot interaction study for a number of reasons. The PER was designed to operate in a museum environment under heavy usage for weeks and months at a time, and PERs have been installed in six science centers around the United States.

3. The Personal Exploration Rover

The Personal Exploration Rover (PER) is the third rover designed and built as part of the Personal Rover Project [12]. The goal of this project is to design and build interactive robots capable of educating and inspiring children. The PER was designed as a tool to educate the public about certain aspects of NASA's Mars Exploration Rover (MER) mission. The goals of the PER are to demonstrate to the public that rovers are tools used for doing science and to illustrate the value of on-board rover autonomy.

Physically, the PER is reminiscent of the MER in its overall mechanical design (Fig. 1(a)). The PER is a six-wheeled robot that uses a rocker-bogie suspension system similar to that used on the MERs. The PER is equipped with a camera and range finder mounted on a pan-tilt head as well as an ultraviolet light for conducting simulated scientific testing.



(a) The PER examines a rock.

(b) A docent talks about the PER with two young visitors.

Figure 1. The PER at (a) the National Science Center and (b) the Smithsonian National Air and Space Museum.

The PER museum exhibit consists of a PER deployed inside a simulated Martian environment (the “Mars yard”) complete with several large rocks as “science targets” and an interactive kiosk, equipped with a trackball and a single button. The premise of the exhibit is that visitors will use the robot to search for life within the Mars yard. The robot is able to test for signs of life using a simulated organofluorescence test, in which the robot shines a UV light on a rock. As the robot conducts the test, it sends a picture of the rock back to the kiosk, where visitors look for a “glow” indicating the presence of (simulated) organic material. The reliability and robustness of the PERs combined with their use in museum exhibits around the United States provide an ideal setting for observing and analyzing long-term human-robot interaction.

There are a three different groups of individuals who have had interactions with the PERs since the PER project began. These are the creators of the PERs at Carnegie Mellon University, museum employees at the PER installation sites, and the museum visitors who use the PER exhibit. Reference [13] is a study of how visitors interact with and react to the PER exhibit, but these interactions rarely last more than several minutes. Museum employees, including administrators, explainers, and technical support people, were chosen to be the focus of this study due to their regular interactions with the PERs over a period of months. These interactions include setting up the PERs at the start of the day, changing their batteries, diagnosing and repairing problems, and talking about the PERs and their exhibit to museum visitors (Fig. 1(b)). In addition, museum employees together form a group of naive initial users who will learn over time and develop cognitive models that they initially may not have had. These two characteristics make them a group well-suited for a study of long-term human-robot interaction.

4. Methodology

For this study, the authors’ goal was to develop a methodology that would enable them to answer the following types of questions about employees’ cognitive models of the PER:

- How does the employee’s conception of robot intelligence change over weeks of interaction?

- How do employees anthropomorphize the robot, if at all?
- As employees gain more experience working with the robot, how do their descriptions of its capabilities change?
- How do employees see the connection between the PER and the MER?

In order to answer these questions, the authors conducted periodic interviews with museum employees from December 2003 through June 2004. These open-ended interviews were conducted once before the PER exhibit had been installed, one to two weeks after the exhibit had been installed, approximately one and a half months after installation, and approximately three and a half months after installation. The exact questions asked to employees at each interview varied slightly, but each employee had an equal opportunity to comment on all question topics. Eighteen museum employees at four PER installations were interviewed; of these, only eleven were able to complete three interviews.

After the interviews were transcribed, a coding scheme was designed to reflect the museum employees' thoughts about the robot. The development of a coding scheme for categorizing types of utterances with respect to learning and museums can be found in [14] and [15]. The coding scheme presented here is based upon both the content of the interviews as well as previous related work. The following nine themes are included in the coding scheme, grouped into three major categories. Each of these nine themes contains a number of sub-codes, but for the purposes of this paper the data have been collapsed up to the super-category level. The three major categories and nine content codes are as follows:

1. Technical talk about the PER

    - Capabilities of the robot
      This theme represents comments about what the PER can and cannot do in terms of its physical components, its behaviors, and its kiosk interface.
    - Failures
      This theme is applied to comments about how the robot failed and the ability of employees to diagnose and solve problems.
    - Reliability
      This theme is used to describe comments about the robot's robustness and resistance to failure.
    - Criteria for intelligence
      This theme focuses on what reasons museum employees give for saying that the PER is intelligent or unintelligent.

2. People and the PER

    - Robot anthropomorphization
      This theme encompasses remarks that museum employees make that the PER "wants", "feels", or "knows" something or that employees or visitors are treating the PER as if it were a living being. Previous work on robot anthropomorphization over the short term can be found in [16] and [17].
    - Visitor description
      This theme is used to characterize comments made by museum employees about how visitors are interacting with the exhibit and how they are treated by employees, either as passive or active learners [18].

Distribution of Content Codes

		Interview	
Code	*1*	*2*	*3*
Reliability	1.1%	*7.1%	6.1%
Anthropomorphization	1.1%	*10.1%	18.4%
Intelligence	1.7%	*6.4%	4.3%
Different POV	7.1%	4.0%	0.5%
MER mission	11.1%	8.5%	*4.3%
Role of robot	12.2%	4.1%	0.5%
Capabilities	14.5%	10.9%	13.5%
Failures	17.0%	17.3%	16.7%
Visitor description	34.1%	*31.5%	35.8%

Figure 2. For each interview and content code, the value listed is equal to the ratio of the number of times that that content code was used out of the total number of lines coded. *Indicates a statistically significant change (one-way repeated-measures ANOVA).

3. PER-MER connections
   - Relationship to the MER mission
     This theme is used for comments museum employees make about how the PER is related to the MERs and their mission.
   - The role of a robot
     This theme is used to represent how museum employees perceive the role of the PER and/or the MER; whether it is a tool used by humans or a machine that collaborates with humans.
   - Taking different points of view
     This theme encompasses the museum employees' seeing the world from the perspective of the PER or of a NASA mission scientist. This theme is adapted from the theme of "Identification with technology" as introduced by [9], a study of the educational impact of a course on robotic autonomy.

The interview transcripts were coded according to the procedure used in [9] and [10]. Out of the lines that were eligible for one of the nine thematic codes, 92.6% of the lines were described unambiguously by one of the thematic codes. The high rate of lines that could be unambiguously described by codes supports the validity of this coding scheme and suggests that the scheme fit the data well.

5. Results

All together, the forty-four interviews generated 2,821 lines that were coded according to the scheme described above. The data from the eleven employees who were able to complete three interviews were used to compute matched-sample statistics. This technique of transforming qualitative data into quantitative data is adapted from [11]. The percentages of each theme that were recorded for each interview can be seen in Fig. 2.

Using the data from the eleven museum employees who were interviewed three times, a one-way repeated-measures ANOVA was computed to determine whether or not there were statistically significant differences across time, accounting for individual differences between employees.

The results of this data analysis can be grouped according to the three major content categories described above, with focus on technical language about the PER, interactions between the PER and people, and connections between the PER and the MER.

5.0.1. Technical Talk about the PER

Over the course of the interviews, there were many significant changes in coding frequencies relating to technical talk about the PER robot itself. Employees talked significantly more about the Reliability ($df = 2$, $F = 5.01$, $p < 0.05$) theme and discussed failures more frequently than any other topic besides museum visitors (Fig. 2). In addition, when describing failures, the use of specific technical terms increased significantly ($df = 2$, $F = 6.73$, $p < 0.01$) without a significant increase in the use of general terminology. At the same time, there were no significant changes in talk about the PER's capabilities. This suggests that as the employees became more familiar with the PER, they tended to focus on the robot's actual successes and failures rather than what it was supposed to be capable of achieving.

5.0.2. People and the PER

Talk about anthropomorphization and instances in which museum employees anthropomorphized the PERs also increased significantly ($df = 2$, $F = 11.14$, $p < 0.01$) as did talk about why the PERs are or are not intelligent ($df = 2$, $F = 4.43$, $p < 0.05$); however, Anthropomorphization was the only content code that increased across all three interviews (Fig. 2). In addition, talk about anthropomorphization was significantly positively correlated with talk about visitors, reliability, and intelligence ($N = 44$, $p < 0.05$, $p < 0.01$, and $p < 0.01$, respectively). These results suggest that as employees spent more time with the PER, anthropomorphization was an important part of their cognitive model, one that was related to talk about several other key themes.

5.0.3. PER-MER Connections

Talk relating the PER to the MER became less frequent as the interviews progressed ($df = 2$, $F = 4.46$, $p < 0.05$). This suggests that the focus of the employees' cognitive model tended to shift away from this higher-level concept over time.

6. Conclusion

The fact that there were many significant changes in employees' talk about the PERs between the first and second interviews suggests that regular interaction with a robot for even a couple of weeks has a large impact on a person's cognitive model.

However, as seen in Fig. 2, the only content code that increased across all three interviews was Anthropomorphization. The fact that more content codes did not exhibit this same trend may be due to a number of factors. The PERs themselves do not exhibit a very wide range of behaviors, and so they may not have required employees to spend a significant amount of time interpreting and adapting to them. In addition, unlike the students in the course on robotics autonomy [9], the employees were not challenged to solve a wide variety of problems with the PER on a regular basis. Without the need to apply their knowledge of the PER in a variety of situations, it is possible that employees' cognitive models were not tested in such a way as to cause a greater number of significant changes. It is also possible that employees with different roles had different reactions to the robot, but there is insufficient data to support this kind of analysis.

Based on the changes that were observed in this study, some of the key factors that should be considered when constructing a cognitive model of how people understand robots include:

- A robot's actual failures and successes may be more important than its purported capabilities. In order to aid people in developing accurate cognitive models, it is best to keep robot behavior transparent. Providing this transparency into the robot's successes and failures will allow users to develop the best possible cognitive model, one based on their own experiences rather than on extensive pre-training.
- Anthropomorphism is a broad concept, frequently associated with a number of other concepts, such as reliability. While it is clear that anthropomorphization is an important part of a person's cognitive model of a robot, exactly what role anthropomorphism plays in that model remains an open question.
- Talk about higher-level concepts, such as the idea of robotic intelligence, declined over time but this decrease was matched by an increase in talk about anthropomorphism. This suggests that people may be thinking of the robot less as a machine and more as a collaborator. A quantitative model of long-term human-robot interaction will need to recognize this distinction between "interactive device as robot" and "interactive device as collaborator" as a person moves from one to the other.

To further advance this research on long-term human-robot interaction, a study on the interaction between scientists and a remotely located "robotic astrobiologist" is currently in progress [19]. This kind of attention to understanding people and how they think about robots is crucial in order to develop technologies that will remain useful to people for long periods of time. The next step is to formalize a quantitative model of human-robot interaction. A robot equipped with this model and an adaptive architecture may then be able to generate more fruitful interactions with the humans around it.

Acknowledgements

The authors would like to thank the staff of the Smithsonian Air and Space Museum, the San Francisco Exploratorium, the National Science Center, and the NASA Ames Visitors' Center for all of the time and energy that they have given to this research. This study would not have been possible without their generous support.

References

[1] Fong, T., Nourbakhsh, I., and Dautenhahn, K. (2002). A survey of socially interactive robots: Concepts, design, and applications. Tech. report CMU-RI-TR-02-29, Robotics Institute, Carnegie Mellon University.

[2] Sony Entertainment Robot AIBO. `http://www.aibo.com`.

[3] Kanda, T., Hirano, T., Eaton, D., and Ishiguro, H. (2003). A practical experiment with interactive humanoid robots in a human society. *Third IEEE International Conference on Humanoid Robots*.

[4] Lewis, M., Sycara, K., and Nourbakhsh, I. (2003). Developing a testbed for studying human-robot interaction in urban search and rescue. *Proceedings of the 10th International Conference on Human Computer Interaction*, Crete, Greece.

[5] Wada, K., Shibata, T., Saito, T., and Tanie, K. (2003). Effects of robot assisted activity to elderly people who stay at a health service facility for the aged. *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, Nevada.

[6] Mitsui, T., Shibata, T., Wada, K., and Tanie, K. (2002). Psychophysiological effects by interaction with mental commit robot. *Journal of Robotics and Mechatronics*, 1(1), 20–26.

[7] Burridge, R., Graham, J., Shillcutt, K., Hirsh, R., and Kortenkamp, D. (2003). Experiments with an EVA assistant robot. *Proceedings of the 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space*.

[8] Severinson-Eklundh, K., Huttenrauch, H., and Green, A. (2003). Social and collaborative aspects of interaction with a service robot. *Robotics and Autonomous Systems, Special Issue on Socially Interactive Robots*, 42(3-4).

[9] Nourbakhsh, I., Crowley, K., Wilkinson, K., and Hamner, E. (2003). The educational impact of the Robotic Autonomy mobile robotics course. Tech report CMU-RI-TR-03-18, Robotics Institute, Carnegie Mellon University.

[10] Nourbakhsh, I., Hamner, E., Bernstein, D., Crowley, K., Ayoob, E., Lotter, M., Shelley, S., Hsiu, T., Porter, E., Dunlavey, B., and Clancy, D. (In press). The Personal Exploration Rover: Educational assessment of a robotic exhibit for informal learning venues. *International Journal of Engineering Education, Special Issue on Robotics Education*.

[11] Chi, M. T. H. (1997). Quantifying qualitative analyses of verbal data: A practical guide. *The Journal of the Learning Sciences*, 6(3), 271–315.

[12] Nourbakhsh, I., Hamner, E., Bernstein, D., Crowley, K., Porter, Hsiu, T., Dunlavey, B., E., Ayoob, E., Lotter, M., Shelley, S., Parikh, A., and Clancy, D. (2004). The Personal Exploration Rover: The ground-up design, deployment and educational evaluation of an educational robot for unmediated informal learning sites. Tech. report CMU-RI-TR-04-38, Robotics Institute, Carnegie Mellon University.

[13] Bernstein, D. (2004). Parent, docents, and robots: Examining mediation at a Mars rover exhibit. *Islands of Expertise: An Approach to Exploring the Cognitive Psychology of Childhood*, K. Crowley (Chair). Symposium conducted at the meeting of the Visitor Studies Association, Albuquerque, New Mexico.

[14] Schauble, L., Gleason, M., Lehrer, R., Bartlett, K., Petrosino, A., Allen, A., Clinton, C., Ho, E., Jones, M., Lee, Y., Phillips, J., Seigler, J., and Street J. (2002). Supporting science learning in museums. *Learning Conversations in Museums*, G. Leinhardt, K. Crowley, and K. Knutson (Eds.) Lawrence Erlbaum Associates.

[15] Crowley, K. and Jacobs, M. (2002). Building islands of expertise in everyday family activity. *Learning Conversations in Museums*, G. Leinhardt, K. Crowley, and K. Knutson (Eds.) Lawrence Erlbaum Associates.

[16] Goetz, J., Kiesler, S., and Powers, A. (2003). Matching robot appearance and behavior to tasks to improve human-robot cooperation. *Proceedings of the 12th IEEE Workshop on Robot and Human Interactive Communication (RO-MAN)*, Millbrae, California.

[17] Mori, M. (1982). *The Buddha in the Robot*, Tuttle Publishing.

[18] Paris, S. (2002). *Perspectives on Object-Centered Learning in Museums*, Lawrence Erlbaum Associates.

[19] The Limits of Life in the Atacama (2004). `http://www.frc.ri.cmu.edu/atacama`.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Information Extraction and Machine Learning: Auto-Marking Short Free Text Responses to Science Questions[1]

Jana Z. SUKKARIEH and Stephen G. PULMAN
Computational Linguistics Group
University of Oxford, OX1 2HG, OXFORD, UK

Abstract Traditionally, automatic marking has been restricted to item types such as multiple choice that narrowly constrain how students may respond. More open ended items have generally been considered unsuitable for machine marking because of the difficulty of coping with the myriad ways in which credit-worthy answers may be expressed. Successful automatic marking of free text answers would seem to presuppose an advanced level of performance in automated natural language understanding. However, recent advances in computational linguistics techniques have opened up the possibility of being able to automate the marking of free text responses typed into a computer without having to create systems that fully understand the answers. This paper describes the use of information extraction and machine learning techniques in the marking of short, free text responses of up to around five lines.

Keywords: Auto-marking, (self) assessment tool, information extraction, machine learning, computational linguistics, assessment in the service of learning.

[1] This is a 3-year project funded by the University of Cambridge Local Examination Syndicate.

Introduction

Traditionally, automatic marking has been restricted to item types such as multiple choice that narrowly constrain how students may respond. More open ended items have generally been considered unsuitable for machine marking because of the difficulty of coping with the myriad ways in which credit-worthy answers may be expressed. Moreover, natural languages (NL), English in this case, can be very ambiguous and there are syntactic and semantic computational processing complexities associated with NL. Recent advances in computational linguistics techniques have opened up the possibility of being able to automate the marking of free text responses typed into a computer without having to create systems that fully understand the answers. E-rater developed by the Educational Testing service[2] ([2],[3],[4]) which uses shallow linguistic processing, and the Intelligent Essay Assessor (IEA) developed by Knowledge Analysis Technologies (KAT) [6] which uses latent semantic analysis are examples of (long) essay automatic marking systems. Our aim is to auto-mark free-text responses also, but only short ones of up to 5 lines, for content. E-rater and IEA do not work for such a task. E-rater depends, among other features, on the length of the essay, and IEA cannot tell the difference between “a student wrote an essay” or “an essay wrote a student”. The responses we are dealing with are to factual science questions where there is an objective criteria for right and wrong, for example, the following GCSE biology answer:

Statement of the question	***Marking Scheme (full mark 2)***[3]
Baby polar bears use their mother’s milk to keep them warm. ***Use your biological knowledge to explain how.***	***Any two from*** *Mother’s milk is warm;Milk high energy content / lots of fat /lots of lactose/lots of sugar;Respiration to give energy/heat;Fat used for insulation;*

The system we have developed is experimental, designed to test the accuracy of the methods used. In a real setting, it is unlikely to be used as the sole marker in a high-stakes examination (partly because of legal sensitivies), but rather as an extra (completely consistent, stress and fatigue-proof) marker to check on the performance of human examiners. It could also be used in `formative’ assessment, either for marking tests as a standalone system or as a part of a bigger one with a variety of short free text, multiple choice and graphically based questions. Such a system could be used as part of the learning process; students could use it for independent revision classes, or self-assessment or teachers could use it to free up time spent on marking.

From an initial random sample of data, we could tell that deep linguistic processing techniques were unlikely to work since answers contained a lot of grammatical and spelling mistakes. We were also aware of the limitations of the computational linguistic processing in the face of tackling any of the following:

- **The need for reasoning and making inferences:** Assume a student answers with, *we do not have to wait until Spring* while the marking key is *it can be done at any time*. Similarly, an answer such as *don’t have sperm or egg* will get a 0 incorrectly if there is no mechanism to infer *no fertilisation*.
- **Students tend to use a negation of a negation (for an affirmative):** An answer like *won’t be done only at a specific time* is the same as *will be done at any time*. An answer like *it is not formed from more than one egg and sperm* is the same as saying *formed from one egg and sperm*. This category is merely an instance of the need for more general reasoning and inference outlined above. We have given this case a separate category because here, the wording of the answer is not very different, while in the general case, the wording can be completely different.

[2] http://www.ets.org/research/erater.html

[3] X;Y/D/K;V is equivalent to saying that each of X, [L]={Y, D,K}, and V deserves 1 mark. The student has to write only 2 of these to get the full mark. [L] denotes an equivalence class i.e. Y, D, K are equivalent. If the student writes Y and D s/he will get only 1 mark.

- **Contradictory or inconsistent information:** Other than logical contradiction like *needs fertilisation and does not need fertilisation*, an answer such as *identical twins have the same chromosomes but different DNA* holds inconsistent scientific information that needs to be detected.

After looking carefully at the data we also discovered other issues which will affect assessment of the accuracy of any automated system, namely:

- **Unconventional expression for scientific knowledge:** Examiners sometimes accept unconventional or informal ways of expressing scientific knowledge, for example, 'sperm and egg get together' for 'fertilisation'.
- **Inconsistency across answers:** In some cases, there is inconsistency in marking across answers. Examiners, sometimes, make mistakes.

We expected that information extraction and machine learning techniques were likely candidates for our short answer auto-marking problem since they do not require complete and accurate parsing and are relatively robust in the face of ungrammatical and incomplete sentences. Other systems which tackle the same problem are being developed. The most prominent among them are those developed by Leacock et al. [7] at ETS, Mitchell et al. ([8],[9]) at Intelligent Assessment Technologies and Rosé et al. [12] at Carnegie Mellon University. The 4 systems, ours included, are being developed independently, yet it seems they share similar characteristics. Commercial and resource pressures currently make it impossible to try these different systems on the same data, and so performance comparisons are meaningless: this is a real hindrance to progress in this field. We require, probably, a common test or evaluation suite of questions/answers, in a particular curriculum, that different assessment organizations agree upon to be able to develop and assess these techniques and systems in a controlled and objective way.

In the following section, we briefly remind the reader on what information extraction is, we recapitulate the approaches we used and results described in Sukkarieh et. al ([13],[14]), and we also report on improvements we made since. In section 2, we remind the reader of some machine-learning techniques. We report on our experiments using such techniques and their corresponding results vis-á-vis our auto-marking problem. We conclude by summarising, describing the work-in-progress, and the tasks ahead. Due to the lack of space, we are, unfortunately, going to omit the statements of Biology questions and some interesting example answers.

1. Information Extraction in a Nutshell

Information extraction (IE) techniques pull out pertinent information from a partially syntactically analysed text by applying a set of domain-specific patterns typically built from training data. In general, the information for filling a template may be found within a single sentence, across sequences of sentences, or sometimes in different forms several times within the same short text.

In our auto-marking problem, consider the following training answers:

the egg after fertilisation splits in two	the fertilised egg has divided into two
the egg was fertilised it split in two	One fertilised egg splits into two
one egg fertilised which split into two	1 sperm has fertilized an egg.. that split into two

These are all paraphrases of *It is the same fertilised egg/embryo*, and variants of what is written above could be captured by a pattern like:

singular_det + <fertilised egg> +{<split>; <divide>; <break>} + {in, into} + <two_halves>, where

<fertilised egg> = NP with the content of 'fertilised egg'

singular_det	= {the, one, 1, a, an}
<split>	= {split, splits, splitting, has split, etc.}
<divide>	= {divides, which divide, has gone, being broken...}
<two_halves>	= {two, 2, half, halves}, etc

The pattern basically is all the paraphrases collapsed into one. It is essential that the patterns use the linguistic knowledge we have at the moment, namely, the part-of-speech tags, the noun phrases and verb groups. In our previous example, the requirement that <fertilised egg> is an NP will exclude something like '*one sperm has fertilized an egg*' while accept something like '*an egg which is fertilized ...*' for e.g.
The patterns or templates (we use the terms interchangeably here, although in some applications it makes sense to distinguish them) i.e., the rules that select from each text the information relevant to the task, are built from training data in one of the following ways. In each case we need to devise a language or a grammar to represent these rules. Before describing the methods and the results, we need to state which shallow linguistic properties we are considering and how we 'extract' them.

We have relied on part-of-speech tagging and information on noun phrases and verb groups in the data. We used a Hidden Markov Model part-of-speech (HMM POS) tagger trained on the Penn Treebank corpus, and a Noun Phrase (NP) and Verb Group (VG) finite state machine (FSM) chunker to provide the input to the information extraction pattern matching phase. The NP network was induced from the Penn Treebank, and then tuned by hand. The Verb Group FSM (i.e. the Hallidayean constituent consisting of the verbal cluster without its complements) was written by hand. Shallow analysis makes mistakes, but multiple sources help fill gaps, and in IE this is adequate for most of the time. The general-purpose lexicon contains words with corresponding tags from the British National Corpus and the Wall Street Journal corpus. The domain-specific lexicon is obviously an on-going process.

1.1 Manually-Engineered Patterns

A person writes special knowledge to extract information using grammars and rules. The 3 crucial steps to take in writing extraction rules by hand can be found, among other references on information extraction, in Appelt and Israel (1999). First, all the ways in which the target information is expressed in a given corpus are determined. Second, all the plausible variants of these ways are considered and then written in appropriate patterns. We first describe the grammatical formalism with which we wrote the patterns. A pattern takes the form: *Id :: LHS ==> RHS*, where Id can be a complex term to categorise patterns into groups and subgroups. *LHS* is a *Cat*, where Cat is a (linguistic) category like NP, VG, Det, etc, or one that is user-defined. *RHS* is a list of *Elements,* where possibly each element is followed by a condition and Elements are defined:

```
Element          ==>     Variable | Word/Cat | c(Cat)
                         |?(Element) optional element
                         | (Element; Element) disjunction
                         W(Word)
```

The first step in the pattern matching algorithm is that all patterns are compiled. Afterwards, when an answer arrives for pattern-matching it is first tagged and all phrases (i.e. verb groups-VG and noun phrases-NP) are found. These are then compared with each element of each compiled pattern in turn, until either a complete match is found or all patterns have been tried and no match was found to exist.

The grammar went through stages of improvement ([13],[14]), starting from words, disjunction of words, sequence of words, etc up until the version described above. We also experimented with a different number of answers used for the training data for different questions and, on average, we have achieved 84.5% agreement with examiners scores. Note that the full mark of each question range between 1-4.

Table 1. Results using the manually-written approach

Question	FullMark	Percentage of Agreement
1	2	89.4
2	2	91.8
3	2	84
4	1	91.3
5	2	76.4
6	3	75
7	1	95.6
8	4	75.3
9	2	86.6
Average	----	84

Table 1 shows the results using the last version of the grammar/system on 9 questions in the GCSE biology exams[4]. For each question, we trained on 80% of the positive instances i.e. answers where the mark was > 0 (as should be done), and tested on the positive and negative instances. In total, we had around 200 instances for each question. The following results are the ones we got before we incorporated the spelling corrector into the system and before including rules to avoid some over-generation. Also, we are in the process of fixing a few NP, VG formations and negations of verbs, and all this should make the percentages higher. Due to some inconsistency in the marking, examiners' mistakes and the decisions that we had to make on what we should consider correct or not, independently of a domain expert, 84% average is a good result. Hence, though some of the results look disappointing, the discrepancy between the system and the examiners is not very significant. Furthermore, this agreement is calculated on the whole mark and not on individual sub_marks. This, obviously, makes the result looks worse than what in reality the system's performance is[5]. In the following section, we describe another approach we used for our auto-marking problem.

1.2 Automatic Pattern Learning

The last approach requires skill, much labour, and familiarity with both domain and tools. To save time and labour, various researchers have investigated machine-learning approaches to learn IE patterns. This requires many examples with data to be extracted, and then the use of a suitable learning algorithm to generate candidate IE patterns. One family of methods for learning patterns requires a corpus to be annotated, at least to the extent of indicating which sentences in a text contain the relevant information for particular templates (e.g. [11]). Once annotated, groups of similar sentences can be grouped together, and patterns abstracted from them. This can be done by taking a partial syntactic analysis, and then combining phrases that partially overlap in content, and deriving a more general pattern from them. All that is needed is people familiar with the domain to annotate the text. However, it is still a laborious task. Another family of methods, more often employed for the named entity recognition stage, tries to exploit redundancy in un-annotated data (e.g. [5]). Previously, in [14], we said that we did not want to manually categorise answers into positive or negative instances, since this is a laborious task, and that we will only consider the

[4] We have a demo available for the system.

[5] For more details on the issues that the system faces and the mistakes it makes and their implications please consult the authors.

sample of human marked answers that have effectively been classified into different groups by the mark awarded. However, in practise the noise in these answers was not trivial and, judging from our experience with the manually-written method, this noise can be minimized by annotating the data. After all, if the training data consists of a few hundred answers then it is not such a laborious task, especially if done by a domain expert.

A Supervised Learning or Semi-Automatic Algorithm The following algorithm omits the first 3 steps from the previously described learn-test-modify algorithm in [14]. In these 3 steps we were trying to automate the annotation task. Annotation here is a lightweight activity. Annotating, highlighting or labelling, in our case, simply means going through each student's answer and highlighting parts of the answers that deserve 1 mark. Categories or classes of 1 mark are chosen as this is mainly the guideline in the marking scheme and this is how examiners are advised to do. There is a one-to-one correspondence between 1 part of the marking scheme, 1 mark, and one equivalence class (in our terms). These are separated by semi-colons (;) in the marking scheme. We can replace these steps with, hopefully a more reliable annotation done by a domain expert[6] and we start with the learning process directly. We keep the rest of the steps in the algorithm as they are, namely,

1. *The learning step (generalisation or abstracting over windows):*
   The patterns produced so far are the most-specific ones, i.e. windows of keywords only. We need some generalisation rules that will help us make a transition from a specific to a more general pattern. Starting from what we call a triggering window, the aim is to learn a general pattern that covers or abstracts over several windows. These windows will be marked as 'seen windows'. Once no more generalisation to the pattern at hand can be made to cover any new windows, a new triggering window is considered. The first unseen window will be used as a new triggering window and the process is repeated until all windows are covered (the reader can ask the authors for more details. These are left for a paper of a more technical nature).
2. *Translate the patterns (or rudimentary patterns) learned in step 1 into the syntax required for the marking process (if different syntax is used).*
3. *Expert filtering again for possible patterns.*
4. *Testing on training data. Make additional heuristics on width. Also, add or get rid of some initial keywords.*
5. *Testing on testing data.*

We continue to believe that the best place to look for alternatives, synonyms or similarities is in the students' answers (i.e. the training data). We continue in the process of implementation and testing. A domain expert (someone other than us) is annotating some new training data. We are expecting to report on these results very soon.

2. Machine-Learning Approach

In the previous section, we described how machine-learning techniques can be used in information extraction to learn the patterns. Here, we use machine-learning algorithms to learn the mark. Given a set of training data consisting of positive and negative instances, that is, answers where the marks are 1 or 0, respectively, the algorithm abstracts a model that represents the training data, that is, describing when or when not to give a mark. When faced with a new answer the model is used to give a mark.

Previously in [13], we reported the results we obtained using Nearest Neighbour Classification techniques. In the following, we report our results using two algorithms, namely, decision tree learning and Bayesian learning on the questions shown in the previous section. The first experiments show the results with non-annotated data; we then repeat the experiments with annotated data. As we mentioned earlier, the annotation is very simple: we highlight the part of the answer that deserves 1 mark, meaning that irrelevant material can be ignored. Unfortunately, this does not mean that the training data is noiseless since sometimes annotating the data is less than straightfor-

[6] This does not mean we will not investigate building a tool for annotation since as it will be shown in section 2, annotating the answers has a significant impact on the results.

ward and it can get tricky. However, we try to minimize inconsistency. We used the existing Weka system [15] to conduct our experiments. For lack of space, we will omit the description of the decision tree and Bayesian algorithms and we only report their results. The results reported are on a 10-fold cross validation testing.

For our marking problem, the outcome attribute is well-defined. It is the mark for each question and its values are {0,1, …full_mark}. The input attributes could vary from considering each word to be an attribute or considering deeper linguistic features like a head of a noun phrase or head of a verb group to be an attribute, etc. In the following experiments, each word in the answer was considered to be an attribute.

Furthermore, (Rennie et al. 2003) propose simple heuristic solutions to some problems with naïve classifiers. In Weka, Complement of Naïve Bayes is supposed to be a refinement to the selection process that Naïve Bayes makes when faced with instances where one outcome value has more training data than another. This is true in our case. Hence, we ran our experiments using this algorithm also to see if there was any difference.

Results on Non-Annotated data

We first considered the non-annotated data, that is, the answers given by students, as they are. The first experiment considered the values of the marks to be {0,1, …, full_mark} for each question. The reports of decision tree learning and Bayesian learning are reported in the columns titled DTL1 and NBayes/CNBayes1. The second experiment considered the values of the marks to be either 0 or >0, i.e. we considered two values only. The results are reported in columns DTL2 and NBayes2/CNBayes2. The baseline is the number of answers with the most common mark over the total number of answers multiplied by 100. Obviously, the result of the baseline differs in each experiment only when the sum of the answers with marks greater than 0 exceeds that of those with mark 0. This affected questions 8 and 9 in Table 2 below. Hence, we took the average of both results. It was no surprise that the results of the second experiment were better than the first on questions with the full mark >1. After all, in the second experiment, the algorithm is learning a 0-mark and a symbol for just any mark>0 as opposed to an exact mark in the first. In both experiments, the Naïve Bayes learning algorithm did better than the decision tree learning algorithm and the complement of Naïve Bayes did slightly better or equally well on questions with a full mark of 1, like questions 4 and 7 in the table, while it resulted in a worse performance on questions with full marks >1.

Table 2. Results for Bayesian learning and decision tree learning on non-annotated data

Question	Baseline	*DTL1*	*NBayes/CNBayes1*	DTL2	NBayes/CNBayes2	Stem_DTL2	Stem_Nbayes2
1	69	*73.52*	*73.52 / 66.47*	76.47	81.17 / 73.52	--	--
2	54	*62.01*	*65.92 / 61.45*	62.56	73.18/ 68.15	--	--
3	46	*68.68*	*72.52 / 61.53*	93.4	93.95 / 92.85	--	--
4	58	*69.71*	*75.42 / 76*	69.71	75.42 / 76	--	--
5	54	*60.81*	*66.66 / 53.21*	67.25	73.09 / 73.09	--	--
6	51	*47.95*	*59.18 / 52.04*	67.34	81.63 / 77.55	73.98	80.10
7	73	*88.05*	*88.05 / 88.05*	88.05	88.05 / 88.05	93.03	87.56
8	42 / 57	*41.75*	*43.29 / 37.62*	72.68	70.10/ 69.07	81.44	71.65
9	60 / 70	*61.82*	*67.20 / 62.36*	76.34	79.03 / 76.88	71.51	77.42
Average	60.05	*63.81*	*67.97/62.1*	74.86	79.51/77.3	--	--

Since we were using the words as attributes, we expected that in some cases stemming the words in the answers would improve the results. Hence, we experimented with the answers of 6, 7, 8 and 9 from the list above and the results, after stemming, are reported in the last two columns in Table 2[7]. We notice that whenever there is an improvement, as in question 8, the difference is very little. Stemming does not necessarily make a difference if the attributes/words that could affect the results appear in a root form already. The lack of any difference or worse performance may also be due to the error rate in the stemmer.

Results on Annotated data

We repeated the second experiments with the annotated answers. As we said earlier, annotation means highlighting the part of the answer that deserves 1 mark (if the answer has >=1 mark), so for e.g. if an answer was given a 2 mark then at least two pieces of information should be highlighted and answers with 0 mark stay the same. Obviously, the first experiments could not be conducted since with the annotated answers the mark is either 0 or 1. The baseline for the new data differs and the results are shown in Table 3 below. Again, Naïve Bayes is doing better than the decision tree algorithm. It is worth noting that, in the annotated data, the number of answers whose marks are 0 is less than in the answers whose mark is 1, except for questions 1 and 2. This may have an effect on the results. From getting the worse performance in NBayes2 before Annotation, Question 8 jumps to seventh place. The rest maintained the same position more or less, with question 3 always nearest to the top. Count(Q,1)-Count(Q,0) is highest for questions 8 and 3, where Count(Q,N) is the number of answers whose mark is N. The improvement of performance for question 8 in relation to Count(8,1) was not surprising, since question 8 has a full-mark of 4 and the annotation's role was an attempt at a one-to-one correspondence between an answer and 1 mark. On the other hand, question 1 that was in seventh place in DTL2 before annotation, jumps down to the worst place after annotation. In both cases, namely, NBayes2 and DTL2 after annotation, it seems reasonable to hypothesize that P(Q1) is better than P(Q2) if Count(Q1,1)-Count(Q1,0) >> Count(Q2,1)-Count(Q2,0), where P(Q) is the percentage of agreement for question Q. Furthermore, according to the results of CNBayes in Table 2, we expected that CNBayes will do better on questions 4 and 7. However, it did better on questions 3, 4, 6 and 9. Unfortunately, we cannot see a pattern or a reason.

Table 3. Results for Bayesian learning and decision tree learning on annotated data

Question	Baseline	DTL	NBayes/CNBayes
1	58	74.87	86.69 / 81.28
2	56	75.89	77.43 / 73.33
3	86	90.68	95.69 / 96.77
4	62	79.08	79.59 / 82.65
5	59	81.54	86.26 / 81.97
6	69	85.88	92.19 / 93.99
7	79	88.51	91.06 / 89.78
8	78	94.47	96.31 / 93.94
9	79	85.6	87.12 / 87.87
Average	69.56	84.05	88.03 / 86.85

As they stand, the results of agreement with given marks are encouraging. However, the models that the algorithms are learning are very naïve in the sense that they depend on words only and providing a justification for a student won't be possible. The next step is to try the algorithms on annotated data that has been corrected for spelling and investigate some deeper features or attributes other than words, like the heads of a noun phrase or a verb group or a modifier of the head, etc.

[7] Our thanks to Leonie Ijzereef for the results in the last 2 columns of Table 2.

3. Conclusion

In this paper, we have described the latest refinements and results made on our automarking system described in ([13],[14]), using information extraction techniques where patterns were hand-crafted or semi-automatically learned. We have also described experiments where the problem is reduced to learning a model that describes the training data and use it to mark a new question. At the moment, we are focusing on information-extraction techniques. The results we obtained are encouraging enough to pursue these techniques with deeper linguistic features, especially to be able to associate some confidence measure and some feedback to the student with each answer marked by the system. We are using machine-learning techniques to learn the patterns or at least some rudimentary ones that the knowledge engineer can complete. As we mentioned earlier in section 1.2, this is what we are in the process of doing. Once this is achieved, the next step is to try and build a tool for annotation and also to use some deeper linguistic features or properties or even (partially) parse the students' answers. We have noticed that these answers vary dramatically in their written quality from one group of students to another. For the advanced group, many answers are more grammatical, more complete and with less spelling errors. Hence, we may be able to extract linguistic features deeper than a verb group and a noun group.

Bibliography

[1] Appelt, D. & Israel, D. (1999) Introduction to Information Extraction Technology. IJCAI 99.

[2] Burstein J., Kukich K., Wolff S., Chi Lu, Chodorow M., Braden-Harder L. and Harris M.D. Automated scoring using a hybrid feature identification technique. 1998.

[3] Burstein J., Kukich K., Wolff S., Chi Lu, Chodorow M., Braden-Harder L. and Harris M.D. Computer analysis of essays. In *NCME Symposium on Automated Scoring*, 1998.

[4] Burstein J., Leacock C. and Swartz R. Automated evaluation of essays and short answers. In *5th International Computer Assisted Assessment Conference*. 2001

[5] Collins, M. and Singer, Y. (1999) Unsupervised models for named entity classification. Proceedings Joint SIGDAT Conference on Empirical Methods in NLP & Very Large Corpora.

[6] Foltz P.W., Laham D. and Landauer T.K. Automated essay scoring: Applications to educational technology. 2003. http://www-psych.nmsu.edu/~pfoltz/reprints/Edmedia99.html. Reprint.

[7] Leacock, C. and Chodorow, M. (2003) C-rater: Automated Scoring of Short-Answer Questions. Computers and Humanities 37:4.

[8] Mitchell, T. Russel, T. Broomhead, P. and Aldridge, N. (2002) Towards robust computerized marking of free-text responses. In 6th International Computer Aided Assessment Conference.

[9] Mitchell, T. Russel, T. Broomhead, P. and Aldridge, N. (2003) Computerized marking of short-answer free-text responses. In 29th annual conference of the International Association for Educational Assessment (IAEA), Manchester, UK.

[10] Rennie, J.D.M., Shih, L., Teevan, J. and Karger, D. (2003) Tackling the Poor Assumptions of Naïve Bayes Text Classifiers. http://haystack.lcs.mit.edu/papers/rennie.icml03.pdf.

[11] Riloff, E. (1993) Automatically constructing a dictionary for information extraction tasks. Proceedings 11th National Conference on Artificial Intelligence, pp. 811-816.

[12] Rose, C. P. Roque, A., Bhembe, D. and VanLehn, K. (2003) A hybrid text classification approach for analysis of student essays. In Building Educational Applications Using NLP.

[13] Sukkarieh, J. Z., Pulman, S. G. and Raikes (2003) N. *Auto-marking: using computational linguistics to score short, free text responses*. In the 29th annual conference of the International Association for Educational Assessment (IAEA), Manchester, UK.

[14] Sukkarieh, J. Z., Pulman, S. G. and Raikes (2004) N. *Auto-marking2: An update on the UCLES-OXFORD University research into using computational linguistics to score short, free text responses*. In the 30th annual conference of the International Association for Educational Assessment (IAEA), Philadelphia, USA.

[15] Witten, I. H. Eibe, F. *Data Mining.* Academic Press 2000.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

A Knowledge Acquisition System for Constraint-based Intelligent Tutoring Systems

Pramuditha Suraweera, Antonija Mitrovic and Brent Martin
Intelligent Computer Tutoring Group
Department of Computer Science, University of Canterbury
Private Bag 4800, Christchurch, New Zealand
{psu16, tanja, brent}@cosc.canterbury.ac.nz

Abstract. Building a domain model consumes a major portion of the time and effort required for building an Intelligent Tutoring System. Past attempts at reducing the knowledge acquisition bottleneck by automating the knowledge acquisition process have focused on procedural tasks. We present CAS (Constraint Acquisition System), an authoring system for automatically acquiring the domain model for non-procedural as well as procedural constraint-based tutoring systems. CAS follows a four-phase approach: building a domain ontology, acquiring syntax constraint directly from it, generating semantic constraints by learning from examples and validating the generated constraints. This paper describes the knowledge acquisition process and reports on results of a preliminary evaluation. The results have been encouraging and further evaluations are planned.

1 Introduction

Numerous empirical studies have shown that Intelligent Tutoring Systems (ITS) are effective tools for education. However, developing an ITS is a labour intensive and time consuming process. A major portion of the development effort is spent on acquiring the domain knowledge that accounts for the intelligence of the system. Our goal is to significantly reduce the time and effort required for building a knowledge base by automating the process.

This paper details the Constraint Acquisition System (CAS), which automatically acquires the required knowledge for ITSs by learning from examples. The knowledge acquisition process consists of four phases, initiated by an expert of the domain describing the domain in terms of an ontology. Secondly, syntax constraints are automatically generated by analysing the ontology. Semantic constraints are generated in the third phase from problems and solutions provided by the author. Finally, the generated constraints are validated with the assistance of the author.

The remainder of the paper is initiated by a brief introduction to Constraint-based modelling, the student modelling technique focused in this research, and a brief overview of related research. We then present a detailed description of CAS, including its architecture and a description of the knowledge acquisition process. Finally, conclusions and future work is outlined.

2 Related work

Constraint based modelling (CBM) [6] is a student modelling approach that somewhat eases the knowledge acquisition bottleneck by using a more abstract representation of the domain compared to other commonly used approaches [5]. However, building constraint sets still remains a major challenge. Our goal is to significantly reduce the time and effort required for acquiring the domain knowledge for CBM tutors by automating the knowledge acquisition process. Unlike other automated knowledge acquisition systems, we aim to produce a system that has the ability to acquire knowledge for non-procedural, as well as procedural, domains.

Existing systems for automated knowledge acquisition have focused on acquiring procedural knowledge in simulated environments or highly restrictive environments. KnoMic [10] is a learning-by-observation system for acquiring procedural knowledge in a simulated environment. It generates the domain model by generalising recorded domain experts' traces. Koedinger et al have constructed a set of authoring tools that enable non AI experts to develop cognitive tutors. They allow domain experts to create "Pseudo tutors" which contain a hard coded domain model specific to the problems demonstrated by the expert [3]. Research has also been conducted to generalise the domain model of "Pseudo tutors" by using machine learning techniques [2].

Most existing systems focus on acquiring procedural knowledge by recording the domain expert's actions and generalising recorded traces using machine learning algorithms. Although these systems appear well suited to tasks where goals are achieved by performing a set of steps in a specific order, they fail to acquire knowledge for non-procedural domains, i.e. where problem-solving requires complex, non-deterministic actions in no particular order. Our goal is to develop an authoring system that can acquire procedural as well as declarative knowledge.

The domain model for CBM tutors [7] consists of a set of constraints, which are used to identify errors in student solutions. In CBM knowledge is modelled by a set of constraints that identify the set of correct solutions from the set of all possible student inputs. CBM represents knowledge as a set of ordered pairs of relevance and satisfaction conditions. The relevance condition identifies the states in which the represented concept is relevant, while the satisfaction condition identifies the subset of the relevant states in which the concept has been successfully applied.

3 Constraint Authoring System

The proposed system is an extension of WETAS [4], a web-based tutoring shell that facilitates building constraint-based tutors. WETAS provides all the domain-independent components for a text-based ITS, including the user interface, pedagogical module and student modeller. The pedagogical module makes decisions based on the student model regarding problem/feedback generation, and the student modeller evaluates student solutions by comparing them to the domain model and updates the student model. The main limitation of WETAS is its lack of support for authoring the domain model.

As WETAS does not provide any assistance for developing the knowledge base, typically a knowledge base is composed using a text editor. Although the flexibility of a text editor may be adequate for knowledge engineers, novices tend to be overwhelmed by the task. The goal of CAS (Constraint Authoring System) is to reduce the complexity of the task by automating the constraint acquisition process. As a consequence the time and effort required for building constraint bases should reduce dramatically.

CAS consists of an ontology workspace, ontology checker, problem/solution manager, syntax and semantic constraint generators, and constraint validation as depicted in Figure 1. During the initial phase, the domain expert develops an ontology of the domain in the ontology workspace. This is then evaluated by the ontology checker, and the result is stored in the ontology repository.

The syntax constraints generator analyses the completed ontology and generates syntax constraints directly from it. These constraints are generated from the restrictions on attributes and relationships specified in the ontology. The resulting constraints are stored in the syntax constraints repository.

CAS induces semantic constraints during the third phase by learning from sample problems and their solutions. Prior to entering problems and sample solutions, the domain expert specifies the representation for solutions. This is a decomposition of the solution into components consisting of a list of instances of concepts. For example, an algebraic equation consists of a list of terms in the left hand and a list of terms in the right hand side.

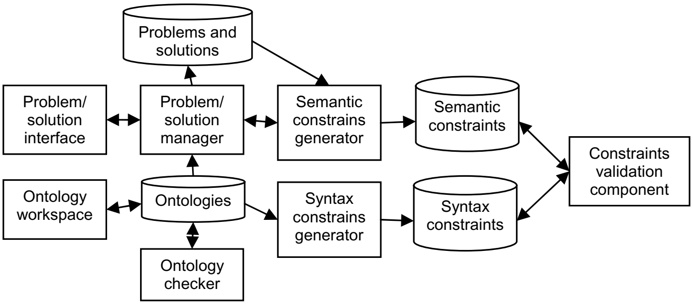

Problems and solutions
Problem/ solution interface
Problem/ solution manager
Semantic constrains generator
Semantic constraints
Constraints validation component
Ontology workspace
Ontologies
Syntax constrains generator
Syntax constraints
Ontology checker

Figure 1: Architecture of the constraint-acquisition system

The final phase involves ensuring the validity of the generated constraints. During this phase the system generates examples to be validated by the author. In situations where the author's validation conflicts with the system's evaluation according to the domain model, the author is requested to provide further examples to illustrate the rationale behind the conflict. The new examples are then used to resolve the conflicts, and may also lead to the generation of new constraints.

3.1 Modelling the domain's ontology

Domain ontologies play a central role in the knowledge acquisition process of the constraint authoring system [9]. A preliminary study conducted to evaluate the role of ontologies in manually composing a constraint base showed that constructing a domain ontology assisted the composition of the constraints [8]. The study showed that ontologies help organise constraints into meaningful categories. This enables the author to visualise the constraint set and to reflect on the domain, assisting them to create more complete constraint bases.

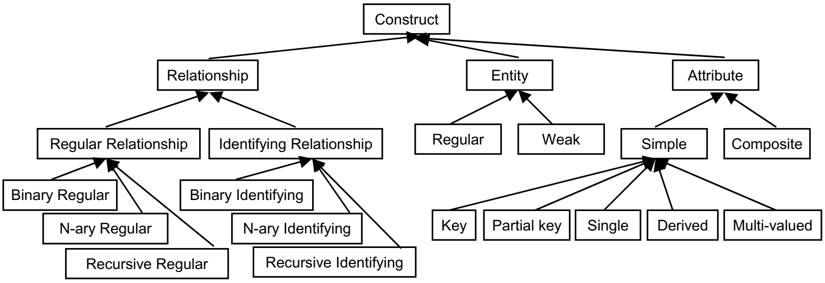


Figure 2: Ontology for ER modelling domain

An ontology describes the domain by identifying important concepts and relationships between them. It outlines the hierarchical structure of the domain in terms of sub- and super-concepts. CAS contains an ontology workspace for modelling an ontology of the domain. An example ontology for Entity Relationship Modelling is depicted in Figure 2. The root node, *Construct,* is the most general concept, of which *Relationship*, *Entity* and *Attribute* are sub-concepts. *Relationship* is further specialised into *Regular* and *Identifying*, which are the two possible types of relationships, and so on.

As syntax constraints are generated directly from the ontology, it is imperative that all relationships are correct. The ontology checker verifies that the relationships between concepts are correct by engaging the user in a dialog. The author is presented with lists of specialisations of concepts involved in a relationship and is asked to label the specialisations that are incorrect. For example, consider a relationship between *Binary identifying relationship* and *Attribute*. CAS asks whether all of the specialisations of *attribute* (*key, partial key, single-valued* etc) can participate in this relationship. The user indicates that *key* and *partial key* attributes cannot be used in this relationship. CAS therefore replaces the original relationship with specialised relationships between *Binary identifying relationship* and the nodes *single-valued*, *multi-valued* and *derived*.

Ontologies are internally represented in XML. We have defined set of XML tags specifically for this project, which can be easily be transformed to a standard ontology representation form such as DAML [1]. The XML representation also includes positional and dimensional details of each concept for regenerating the layout of concepts in the ontology.

3.2 Syntax Constraint Generation

An ontology contains much of information about the syntax of the domain: information about domain concepts; the domains (i.e. possible values) of their properties; restrictions on how concepts participate in relationships. Restrictions on a property can be specified in terms of whether its value has to be unique or whether it has to contain a certain value. Similarly, restrictions on the participation in relationships can also be specified in terms of minimum and maximum cardinality.

The syntax constraints generator analyses the ontology and generates constraints from all the restrictions specified on properties and relationships. For example, consider the *owner* relationship between *Binary identifying relationship* and *Regular entity* from the ontology in Figure 2, which has a minimum cardinality of 1. This restriction specifies that each *Binary identifying relationship* has to have at least one *Regular entity* participating as the *owner*, and can be translated to a constraint that asserts that each *Identifying relationship* found in a solution has to have at least one *Regular entity* as its *owner*.

To evaluate the syntax constraints generator, we ran it over the ER ontology in Figure 2. It produced a total of 49 syntax constraints, covering all the syntax constraints that were manually developed for KERMIT [7], an existing constraint-based tutor for ER modelling. The generated constraint set was more specific than the constraints found in KERMIT, i.e. in some cases several constraints generated by CAS would be required to identify the problem states identified by a single constraint in KERMIT. This may mean that the set of generated constraints would be more effective for an ITS, since they would provide feedback that is more specific to a single problem state. However, it is also possible that they would be overly specific.

We also experimented with basic algebraic equations, a domain significantly different to ER modelling. The ontology for algebraic equations included only four basic operations: addition, subtraction, multiplication and division. The syntax constraints generator produced three constraints from an ontology composed for this domain, including constraints that ensure whenever an opening parenthesis is used there should be a corresponding closing parenthesis, a constant should contain a plus or minus symbol as its sign, and a constant's value should be greater than or equal to 0. Because basic algebraic expressions have very little syntax restrictions, three constraints are sufficient to impose the basic syntax rules.

3.3 Semantic Constraint Generation

Semantic constraints are generated by a machine learning algorithm that learns from examples. The author is required to provide several problems, with a set of correct solutions for each depicting different ways of solving it. A solution is composed by populating each of its components by adding instances of concepts, which ensures that a solution strictly adheres to the domain ontology. Alternate solutions, which depict alternate ways of solving the problem, are composed by modifying the first solution. The author can transform the first solution into the desired alternative by adding, editing or dropping elements. This reduces the amount of effort required for composing alternate solutions, as most alternatives are similar. It also enables the system to correctly identify matching elements in two alternate solutions.

The algorithm generates semantic constraints by analysing pairs of solutions to identify similarities and differences between them. The constraints generated from a pair of solutions contribute towards either generalising or specialising constraints in the main constraint base. The detailed algorithm is given in Figure 3.

```
a. For each problem $P_i$
b. For each pair of solutions $S_i$ & $S_j$
     a. Generate a set of new constraints N
     b. Evaluate each constraint $CB_i$ in main constraint base, CB, against $S_i$ & $S_{j,}$
          If $CB_i$ is violated, generalise or specialise $CB_i$ to satisfy $S_i$ & $S_j$
     c. Evaluate each constraint $N_i$ in set N against each previously analysed pair of solutions $S_x$ & $S_y$ for each previously analysed problem $P_z$,
          If $N_i$ is violated, generalise or specialise $CB_i$ to satisfy $S_x$ & $S_y$
     d. Add constraints in N that were not involved in generalisation or specialisation to CB
```

Figure 3: Semantic constraint generation algorithm

The constraint learning algorithm focuses on a single problem at a time. Constraints are generated by comparing one solution to another of the same problem, where all permutations of solution pairs, including solutions compared to themselves, are analysed. Each solution pair is evaluated against all constraints in the main constraint base. Any that are violated are either specialised to be irrelevant for the particular pair of solutions, or generalised to satisfy that pair of solutions. Once no constraint in the main constraint base is violated by the solution pair, the newly generated set of constraints is evaluated against all previously analysed pairs of solutions. The violated constraints from this new set are also either specialised or generalised in order to be satisfied. Finally, constraints in the new set that are not found in the main constraint base are added to the constraint base.

```
1. Treat $S_i$ as the ideal solution (IS) and $S_j$ as the student solution (SS)
2. For each element A in the IS
     a. Generate a constraint that asserts that if IS contains the element A, SS should contain a matching element
     b. For each relationship that element is involved with,
          Generate constraints that ensures that the relationship holds between the corresponding elements of the SS
3. Generalise the properties of similar constraints by introducing variables or wild cards
```

Figure 4: Algorithm for generating constraints from a pair of solutions

New constraints are generated from a pair of solutions following the algorithm outlined in Figure 4. It treats one solution as the ideal solution and the other as the student solution. A constraint is generated for each element in the ideal solution, asserting that if the ideal solution contains the particular element, the student solution should also contain the matching element.

E.g. Relevance: IS.Entities has a Regular entity
Satisfaction: SS.Entities has a Regular entity

In addition, three constraints are generated for each relationship that an element participates with. Two constraints ensure that a matching element exists in SS for each of the two elements of IS participating in the relationship. The third constraint ensures that the relationship holds between the two corresponding elements of SS.

E.g. 1. Relevance: IS.Entities has a Regular entity
AND IS.Attributes has a Key
AND SS.Entities has a Regular entity
AND IS Regular entity is in *key-attribute* with Key
AND IS Key is in *belong to* with Regular entity
Satisfaction: SS.Attributes has a Key

2. Relevance: IS.Entities has a Regular entity
AND IS.Attributes has a Key
AND SS.Attributes has a Key
AND IS Regular entity is in *key-attribute* with Key
AND IS Key is in *belong to* with Regular entity
Satisfaction: SS.Entities has a Regular entity

3. Relevance: IS.Entities has a Regular entity
AND IS.Attributes has a Key
AND SS.Entities has a Regular entity
AND SS.Attributes has a Key
AND IS Regular entity is in *key-attribute* with Key
AND IS Key is in *belong to* with Regular entity
Satisfaction: SS Regular entity is in *key-attribute* with Key
AND SS Key is in *belong to* with Regular entity

```
a. If constraint set, C-set that does not contain violated constraint V, has a similar but a more
   restrictive constraint C then replace V with C and exit.
b. If C-set has a constraint C that has the same relevance condition but different satisfaction
   condition to V,
      Add the satisfaction condition of C as a disjunctive test to the satisfaction of V, remove C
      from C-set and exit
c. Find a solution $S_k$ that satisfies constraint V
d. If a matching element can be found in $S_j$ for each element in $S_k$ that appears in the satisfaction condition,
      Generalise satisfaction of V to include the matching elements as a new test with a disjunction and exit
e. Restrict the relevance condition of V to be irrelevant for solution pair $S_i$ & $S_j$, by adding a
   new test to the relevance signifying the difference and exit
f. Drop constraint
```

Figure 5: Algorithm for generalising or specialising violated constraints

The constraints that get violated during the evaluation stage are either specialised or generalised according to the algorithm outlined in Figure 5. It deals with two sets of constraints (C-set): the new set of constraints generated by a pair of solutions and the main constraint base. The algorithm remedies each violated constraint individually by either specialising it or generalising it. If the constraint cannot be resolved, it is labelled as an incorrect constraint and the system ensures that it does not get generated in the future.

The semantic constraints generator of CAS produced a total of 135 constraints for the domain of ER modelling using the ontology in Figure 2 and six problems. The problems supplied to the system were simple and similar to the basic problems offered by KERMIT. Each problem focused on a set of ER modelling constructs and contained at least two solutions that exemplified alternate ways of solving the problem. The solutions were selected that maximised the differences between them. The differences between most solutions were small because ER modelling is a domain that does not have vastly different solutions. However, problems that can be solved in different ways consisted of significantly different solutions.

The generated constraints covered 85% of the 125 constraints found in KERMIT's constraint-base, which was built entirely manually and has proven to be effective. After further analysing the generated constraints, it was evident that the reason for not generating most of the missing constraints was due to a lack of examples. 85% coverage is very encouraging, considering the small set of sample problems and solutions. It is likely that providing further sample problems and solutions to CAS would increase the completeness of the generated domain model. Although the problems and solutions were specifically chosen to improve the system's effectiveness in producing semantic constraints, we assume that a domain expert would also have the ability to select good problems and provide solutions that show different ways of solving a problem. Moreover, the validation phase, which is yet to be completed, would also produce constraints with the assistance of the domain expert.

CAS also produced some modifications to existing constraints found in KERMIT, which improved the system's ability to handle alternate solutions. For example, although the constraints in KERMIT allowed weak entities to be modelled as composite multivalued attributes, in KERMIT the attributes of weak entities were required to be of the same type as the ideal solutions. However CAS correctly identified that when a weak entity is represented as a composite multivalued attribute, the partial key of the weak entity has to be modelled as simple attributes of the composite attribute. Furthermore, the identifying relationship essential for the weak entity becomes obsolete. These two examples illustrate how CAS improved upon the original domain model of KERMIT.

We also evaluated the algorithm in the domain of algebraic equations. The task involved specifying an equation for the given textual description. As an example, consider the problem "Tom went to the shop to buy two loafs of bread, he gave the shopkeeper a \$5 note and was given \$1 as change. Write an expression to find the price of a loaf of bread using x to represent the price". It can be represented as $2x + 1 = 5$ or $2x = 5 – 1$. In order to avoid the need for a problem solver, the answers were restricted to not include any simplified equations. For example the solution "x = 2" would not be accepted because it is simplified.

```
a)  Relevance: IS LHS has a Constant (?Var1)
    Satisfaction: SS LHS has a Constant (?Var1)
        or SS RHS has a Constant (?Var1)

b)  Relevance: IS RHS has a +
    Satisfaction: SS LHS has a –
        or SS RHS has a +

c)  Relevance: IS RHS has a Constant(?Var1)
        and IS RHS has a  –
        and SS LHS has a Constant(?Var1)
        and SS LHS has a +
        and IS Constant (?Var1) is in Associated-operator with –
    Satisfaction: SS Constant (?Var1) is in Associated-operator with +
```

Figure 6: Sample constraints generated for Algebra

The system was given five problems and their solutions involving addition, subtraction, division and multiplication for learning semantic constraints. Each problem contained three or four alternate solutions. CAS produced a total of 80 constraints. Although the completeness of the generated constraints is yet to be formally evaluated, a preliminary assessment revealed that the generated constraints are able to identify correct solutions and point out many errors. Some generated constraints are shown in Figure 6. An algebraic equation consists of two parts: a left hand side (LHS) and a right hand side (RHS). Constraint *a* in Figure 6 specifies that for each constant found in the LHS of the Ideal solution (IS), there has to be an equal constant in either the LHS or the student solution (SS) or the RHS. Similarly, constraint *b* specifies that an addition symbol found in the RHS of the IS should exist in the SS as either an addition symbol in the same side or a subtraction in the opposite side. Constraint *c* ensures the existence of the relationship between the operators and the constants. Thus, a constant in the RHS of the IS with a subtraction attached to it, can appear as a constant with addition attached to it in the LHS of the SS.

4 Conclusions and Future work

We provided an overview of CAS, an authoring system that automatically acquires the constraints required for building constraint-based Intelligent Tutoring Systems. It follows a four-stage process: modelling a domain ontology, extracting syntax constraints from the ontology, generating semantic constraints and finally validating the generated constraints.

We undertook a preliminary evaluation in two domains: ER modelling and algebra word problems. The domain model generated by CAS for ER modelling covered all syntax constraints and 85% of the semantic constraints found in KERMIT [7] and unearthed some discrepancies in KERMIT's constraint base. The results are encouraging, since the constraints were produced by analysing only 6 problems. CAS was also used to produce constraints for the domain of algebraic word problems. Although the generated constraints have not been formally analysed for their completeness, it is encouraging that CAS is able to handle two vastly different domains.

Currently the first three phases of the constraints acquisition process have been completed. We are currently developing the constraint validation component, which would also contribute towards increasing the quality of the generated constraint base. We also will be enhancing the ontology workspace of CAS to handle procedural domains. Finally, the effectiveness of CAS and its ability to scale to domains with large constraint bases has to be empirically evaluated in a wide range of domains.

References

[1] DAML. DARPA Agent Markup Language, http://www.daml.org.

[2] Jarvis, M., Nuzzo-Jones, G. and Heffernan, N., *Applying Machine Learning Techniques to Rule Generation in Intelligent Tutoring Systems*. In: Lester, J., et al. (eds.) Proc. ITS 2004, Maceio, Brazil, Springer, pp. 541-553, 2004.

[3] Koedinger, K., et al., *Openning the Door to Non-programmers: Authoring Intelligent Tutor Behavior by Demonstration*. In: Lester, J., et al. (eds.) Proc. ITS 2004, Maceio, Brazil, Springer, pp. 162-174, 2004.

[4] Martin, B. and Mitrovic, A., *WETAS: a Web-Based Authoring System for Constraint-Based ITS*. Proc. 2nd Int. Conf on Adaptive Hypermedia and Adaptive Web-based Systems AH 2002, Malaga, Spain, LCNS, pp. 543-546, 2002.

[5] Mitrovic, A., Koedinger, K. and Martin, B., *A comparative analysis of cognitive tutoring and constraint-based modeling*. In: Brusilovsky, P., et al. (eds.) Proc. 9th International conference on User Modelling UM2003, Pittsburgh, USA, Springer-Verlag, pp. 313-322, 2003.

[6] Ohlsson, S., *Constraint-based Student Modelling*. Proc. Student Modelling: the Key to Individualized Knowledge-based Instruction, Berlin, Springer-Verlag, pp. 167-189, 1994.

[7] Suraweera, P. and Mitrovic, A. *An Intelligent Tutoring System for Entity Relationship Modelling*. Int. J. Artificial Intelligence in Education, vol 14 (3,4), 2004, pp. 375-417.

[8] Suraweera, P., Mitrovic, A. and Martin, B., *The role of domain ontology in knowledge acquisition for ITSs*. In: Lester, J., et al. (eds.) Proc. Intelligent Tutoring Systems 2004, Maceio, Brazil, Springer, pp. 207-216, 2004.

[9] Suraweera, P., Mitrovic, A. and Martin, B., *The use of ontologies in ITS domain knowledge authoring*. In: Mostow, J. and Tedesco, P. (eds.) Proc. 2nd Int. 2nd International Workshop on Applications of Semantic Web for E-learning SWEL'04, ITS2004, Maceio, Brazil, pp. 41-49, 2004.

[10] van Lent, M. and Laird, J.E., *Learning Procedural Knowledge through Observation*. Proc. International conference on Knowledge capture, pp. 179-186, 2001.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Computer Games as Intelligent Learning Environments: A River Ecosystem Adventure

Jason TAN, Chris BEERS, Ruchi GUPTA, and Gautam BISWAS
Dept. of EECS and ISIS, Vanderbilt University
Nashville, TN, 37235, USA
{jason.tan, chris.beers, ruchi.gupta, gautam.biswas}@vanderbilt.edu

Abstract. Our goal in this work has been to bring together the entertaining and flow characteristics of video game environments with proven learning theories to advance the state of the art in intelligent learning environments. We have designed and implemented an educational game, a river adventure. The adventure game design integrates the Neverwinter Nights game engine with our teachable agents system, Betty's Brain. The implementation links the game interface and the game engine with the existing Betty's Brain system and the river ecosystem simulation using a controller written in Java. After preliminary testing, we will run a complete study with the system in a middle school classroom in Fall 2005.

Keywords: educational games, video game engines, teachable agents, intelligent learning environments

Introduction

Historically, video and computer games have been deemed counterproductive to education [1]. Some educators, parents, and researchers believe that video games take away focus from classroom lessons and homework, stifle creative thinking, and promote unhealthy individualistic attitudes [1,2]. But many children find these games so entertaining that they seem to play them nonstop until they are forced to do something else. As a result, computer and video games have become a huge industry with 2001 sales exceeding $6 billion in the United States alone [3].

Research into the effects of video games on behavior has shown that not all of the criticism is justified [3]. State of the art video games provide immersive and exciting virtual worlds for players. They use challenge, fantasy, and curiosity to engage attention. Interactive stories provide context, motivation, and clear goal structures for problem solving in the game environment. Researchers who study game behavior have determined that they place users in *flow states*, i.e., "state[s] of optimal experience, whereby a person is so engaged in activity that self-consciousness disappears, sense of time is lost, and the person engages in complex, goal-directed activity not for external rewards, but simply for the exhilaration of doing." [4]

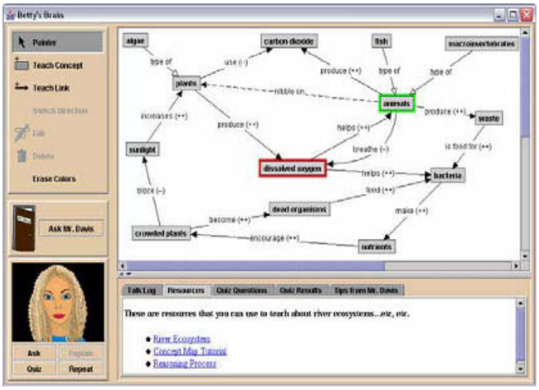
The Sims (SimCity, SimEarth, etc.), Carmen Sandiego, Pirates, and Civilization are examples of popular games with useful educational content [3]. However, the negative baggage that has accompanied video games has curtailed the use of advanced game platforms in learning environments. Traditional educational games tend to be mediocre drill and practice environments (e.g., MathBlaster, Reader Rabbit, and Knowledge Munchers) [5]. In a recent attempt to harness the advantages of a video game framework for learning 3D mathematical functions, a group of researchers concluded that doing so was a mistake. "By telling the students beforehand that they were going to be using software that was game-like in nature, we set the [computer learning environment] up to compete against commercial video games. As can be seen by the intense competition present in the commercial video game market, the students' high expectations are difficult to meet." [6].

What would we gain by stepping up and facing the challenge of meeting the high expectations? Integrating the "flow" feature of video games with proven learning theories to design learning environments has tremendous potential. Our goal is to develop learning environments that combine the best features of game environments and learning theories. The idea is to motivate students to learn by challenging them to solve realistic problems, and exploit animation and immersive characteristics of game environments to create the "flow" needed to keep the students engaged in solving progressively more complex learning tasks.

In previous work, we have developed Betty's Brain, a teachable agent that combines learning by teaching with self-regulated mentoring to promote deep learning and understanding [7]. Experiments in fifth grade science classrooms demonstrated that students who taught Betty showed deep understanding of the content material and developed far transfer capabilities [8]. Students also showed a lot of enthusiasm by teaching Betty beyond the time allocated, and by putting greater effort into reading resources so that they could teach Betty better.

A study of game genres [9] has led us to adopt an adventure game framework for extending the Betty's Brain system. We have designed a game environment, where Betty and the student team up and embark on a river adventure to solve a number of river ecosystem problems. Their progress in the game is a function of how well Betty has been taught about the domain, and how proficient they are in implementing an inquiry process that includes collecting relevant evidence, forming hypotheses, and then carrying on further investigations to support and refine the hypotheses. This paper discusses the interactive story that describes the game structure and the problem episodes.

1. Learning by Teaching: The Betty's Brain System



These are resources that you can use to teach about river ecosystems...etc, etc.

- River Ecosystems
- Concept Map Tutorial
- Reasoning Process

Figure 1. Betty's Brain Interface

Our work is based on the intuitively compelling paradigm, *learning by teaching*, which states that the process of teaching helps one learn with deeper understanding [7]. The teacher's conceptual organization of domain concepts becomes more refined while communicating ideas, reflecting on feedback, and by observing and analyzing the students' performance. We have designed a computer-based system, Betty's Brain, shown in Fig. 1, where students explicitly teach a computer agent named Betty [10]. The system has been used to teach middle school students about interdependence and balance in river ecosystems. Three activities, *teach, query, and quiz*, model student-teacher interactions. In the teach mode, students teach Betty by constructing a concept map using a graphical drag and drop interface. In the query mode, students can query Betty about the concepts they have taught her. Betty uses qualitative reasoning mechanisms to reason with the concept map. When asked, she uses a combination of text, speech, and animation to provide a detailed explanation of how she derived her answer. In the quiz mode, students observe how Betty performs on pre-scripted questions. This feedback tells students how well they have taught Betty, which in turn helps them to reflect on how well they have learned the information themselves. To extend students' understanding of interdependence to balance in river ecosystems, we introduced temporal structures and corresponding reasoning mechanisms into Betty's concept map representation. In the extended framework, students teach Betty to identify cycles (these correspond to feedback loops in dynamic processes) in the concept map and assign time information to each cycle. Betty can now answer questions like, "*If macroinvertebrates increase what happens to waste in two weeks?*" A number of experimental studies in fifth grade science classrooms have demonstrated the effectiveness of the system [8].

The river ecosystem simulation, with its visual interface, provides students with a window to real world ecosystems, and helps them learn about dynamic processes. Different scenarios that include the river ecosystem in balance and out of balance illustrate cyclic processes and their periods, and that large changes (such as dumping of waste) can cause large fluctuations in entities, which leads to eventual collapse of the ecosystem. The simulation interface uses animation, graphs, and qualitative representations to show the dynamic relations between entities in an easy to understand format. Studies with high school students have shown that the simulation helps them gain a better understanding of the dynamics of river ecosystems [11]. This has motivated us to extend the system further and build a simulation based game environment to create an entertaining exploratory environment for learning.

2. Game Environment Design

Good learning environments must help students develop life-long learning and problem solving skills [12]. Betty's Brain, through the Mentor feedback and Betty's interactions with the student-teacher, incorporates metacognitive strategies that focus on self-regulated learning [8]. In extending the system to the game environment, we hope to teach general strategies that help students apply what they have learnt to problem solving tasks. The River Ecosystem Adventure, through cycles of problem presentation, learning, teaching, and problem solving, is designed to provide a continual flow of events that should engage students and richly enhance their learning experience (see Fig. 2). Students are given opportunities to question, hypothesize, investigate, analyze, model, and evaluate; the six phases of the scientific inquiry cycle not only help students acquire new knowledge, but develop metacognitive strategies that lead to generalized problem solving skills and transfer [13].

The game environment is set in a world where students interact with and solve problems for communities that live along a river. The teachable agent architecture is incorporated into the game environment. The student player has a

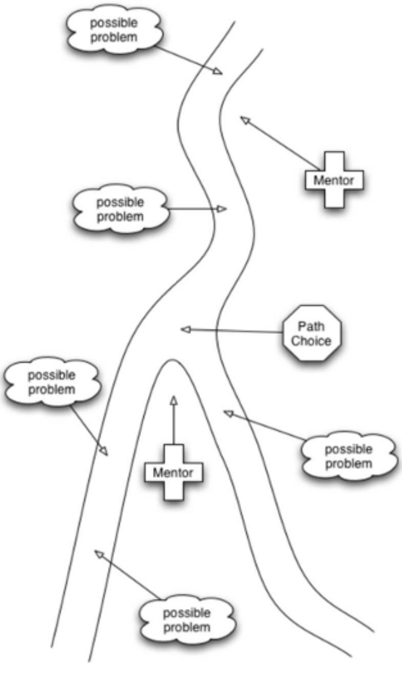


Figure 2. Abstract view of the river

primary "directorial" role in all phases of game play: learning and teaching, experimenting, and problem solving. In the prelude, students are introduced to the game, made familiar with the training academy and the experimental pond, and given information about the ecosystem problems they are likely to encounter on the river adventure. The *learning and teaching phase* mirrors the Betty's Brain environment. The student and Betty come together to prepare for the river adventure in a training academy. Like before, there is an interactive space (the concept map editor) that allows the player to teach Betty using a concept map representation, ask her questions, and get her to take quizzes. Betty presents herself to the student as a disciplined and enthusiastic learner, often egging the student on to teach her more, while suggesting that students follow good self-regulation strategies to become better learners themselves. Betty must pass a set of quizzes to demonstrate that she has sufficient knowledge of the domain before the two can access the next phase of the game. Help is provided in terms of library resources and online documents available in the training academy, and Betty and the student have opportunities to consult a variety of mentor agents who visit the academy.

In the *experiment phase*, Betty and the player accompany a river ranger to a small pond outside of the academy to conduct experiments that are geared toward applying their learnt knowledge to problem solving tasks. The simulation engine drives the pond environment. The ranger suggests problems to solve, and provides help when asked questions. Betty uses her concept map to derive causes for observed outcomes. The ranger analyzes her solutions and provides feedback. If the results are unsatisfactory, the student may return with Betty to the academy for further study and teaching. After they have successfully solved a set of experimental problems, the ranger gives them permission to move on to the adventure phase of the game.

In the *problem-solving phase*, the player and Betty travel to the problem location, where the mayor explains the problem that this part of the river has been experiencing. From this point on, the game enters a real-time simulation as Betty and the student attempt to find a solution to the problem before it is too late. The student gets Betty to approach characters present in the environment, query them, analyze the information provided, and reason with relevant data to formulate problem hypotheses and find possible causes for these hypotheses. The student's responsibility is to determine which pieces of information are relevant to the problem and communicate this information to Betty using a menu-driven interface. Betty reasons with this information to formulate and refine hypotheses using the concept map. If the concept map is correct and sufficient evidence has been collected, Betty generates the correct answer. Otherwise, she may suggest an incorrect cause, or fail to find a solution. An important facet of this process involves Betty explaining to the player why she has selected her solution. Ranger agents appear in the current river location at periodic intervals. They answer queries and provide clues, if asked. If Betty is far from discovering the correct solution, the student can take Betty back to the academy for further learning and teaching. The simulation engine, outlined in section 2, controls the state of the river and data generated in the environment. A screenshot of the game scenario is shown

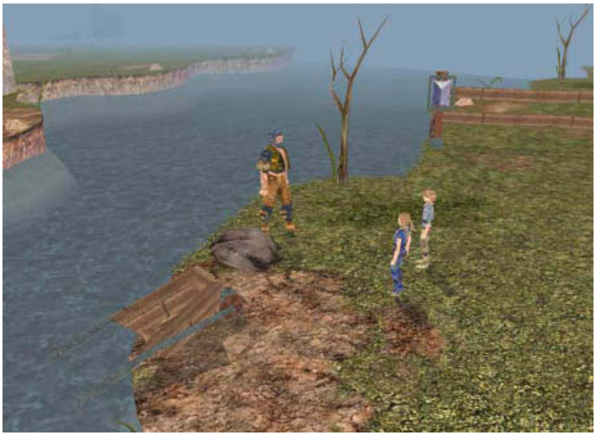


Figure 3. Screenshot of the game

in Fig. 3.

As the simulation clock advances, the problem may get worse and it becomes increasingly urgent for Betty and the student to find a solution. A proposed solution is presented to the mayor, who implements the recommendation. Upon successfully solving and fixing the problem, the team is given a reward. The reward can be used to buy additional learning resources, or conduct more advanced experiments in the pond in preparation for future challenges. The challenges that the students face become more complex in succession.

2.1. Game Engine Selection

In order to accomplish our goal of combining the advantages of current video game technology and an intelligent learning-by-teaching environment, we looked at several adventure/RPG game engines. Most of these game engines provide a variety of scripting tools to control the characters, the dialog structures, and the flow of events in the game. In our work, we felt that a game engine that provides an overhead view of the environment would be most suitable for the student to direct Betty's movements and actions in the world, rather than game engines that provide a first-person point-of-view. This led us to select the Neverwinter Nights game engine from BioWare Corp. [14] as the development environment for this project. The game environment, originally based on the popular game, Dungeons and Dragons, includes the Aurora Toolset, a sophisticated content development toolkit that allows users to create new weapons and monsters, as well as new scenarios and characters using scripted dialogue mechanisms. The toolset has been very successful and has spawned many free user-created expansions.

2.2. Development Process

The Aurora Toolset uses a unique vocabulary for content creation. The adventure is created as a *module* containing all the locations, areas, and characters that make up the game. The module is divided up into regions or *areas* of interest. Each area can take on unique characteristics that contribute to different aspects of the game. The primary character in the game (the student) is the *Player Character (PC)*. A number of other characters not directly under the control of the PC can be included in the adventure. They are called the *Non-Playing Characters (NPC)*. In the River Adventure, Betty has an unusual role of being a NPC who is often controlled by the PC. Each individual problem scenario, the training academy, and the pond define individual areas, and the mentor agents, the rangers, and all other characters in the game environment are NPCs placed in the appropriate areas. Some NPCs can migrate from one area to another.

3. Implementation of the Game Environment

One of the benefits of the Neverwinter Nights game engine is that it can be implemented using a client-server approach. This allows us to separate the simulation engine, Betty's AI-based reasoners, and the other educational aspects of the game from the Neverwinter Nights interface. The underlying system based on the Betty's Brain system with the added functionality (described in Section 3) can then be implemented on the server side, as illustrated in Fig. 4.

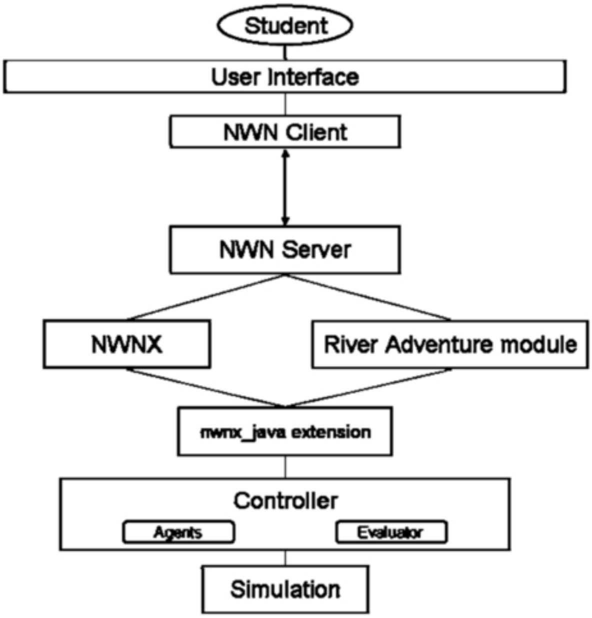


Figure 4. The game environment architecture

A representation of the world is presented to the player by the game engine through the game interface on the client system. The player interacts with the system using a mouse and keyboard to control the movements of his own character and Betty (they move together), click on items of interest (to perform experiments, collect data, check on the concept map, etc.), and to initiate dialog with other NPCs. These define the set of actions that are programmed into the game engine. When students perform an action, it is communicated to the game engine. The game engine controls the visual representation of the world, renders the necessary graphics, and maintains the basic state of the environment and all the characters.

On the server side, the *River Adventure module* describes the location and appearance of each NPC, the details of each area (what buildings and items are present in each scene), how each area connects to other areas, and the overall flow of the game from one level to the next. The Aurora toolset provides a powerful scripting engine used to control the NPC's actions, and other aspects of the module. However, to fully implement the Betty's Brain agent architecture, the river ecosystem simulation, and other more complicated aspects of the system, we utilize the "Neverwinter Nights Extender" (NWNX) [15]. NWNX allows for extensions to the Neverwinter Nights server. In our case, we use the nwnx_java extensions which implements an interface to Java classes and libraries. This allows us to incorporate aspects already implemented in the Betty's Brain system with less effort. The controller and the simulation, implemented in Java, can now be integrated into the River Adventure module. As described in Section 2, the simulation engine uses a state-based mathematical model to keep track of the state of river system as time progresses. Details of this component are presented elsewhere [11], so we do not repeat it here. The rest of this section focuses on the design of the controller, and the updates we made to Betty's reasoning mechanisms to enable her to perform diagnosis tasks.

3.1. The Controller

The controller, made up of the agent architecture and the evaluator, is the core of the intelligent aspects of the game implementation. Additionally, the controller maintains the current state of the game and determines what aspects of the world are accessible to the player. The evaluator assesses the performance of Betty and the student and is used to determine what scaffolding is necessary, as well as maintaining the player's score.

The controller leverages our previous work on multi-agent architecture for learning by teaching systems [8]. Each agent has three primary components: (i) the pattern tracker, (ii) the decision maker, and (iii) the executive. Betty, the mentors and rangers, and all of the significant NPCs in the game world have a corresponding agent within the controller. The pattern tracker monitors the environment, and initiates the decision maker when relevant observable patterns occur. The decision maker takes the input from the pattern tracker and determines what actions the agent should take. Finally, the executive executes these actions, and makes the necessary changes to the environment. Depending on the agent, this could include movement, dialog generation, or a specialized activity, such as making inferences from a concept map or generating help messages. NPC dialogues are generated by retrieving the correct dialog template and modifying it based on the decision maker's output. The controller relays new information resulting from the agents' actions through the nwnx_java plugin to the game module, and also updates the simulation as necessary.

Separate from the agent architecture, the evaluator is the part of the controller that assesses the student's performance and adjusts the game accordingly. The evaluator analyzes the results of the simulation as well as the student's past actions to determine how the game will progress. It takes into account what aspects of the problem the student has yet to complete and sends this information to the game module. The decision makers associated with the mentor agents use this information to determine what level of help the mentors should give the student. If certain aspects of the problem remain unsolved for an extended period of time the mentors can give additional help.

3.2. Betty's extended reasoning mechanisms

Problem solving in the game hinges upon Betty's ability to determine the root cause of a problem given the symptoms and current conditions. Betty's concept map has to be correct and sufficiently complete for her to generate a correct answer. The reasoning mechanism in the existing Betty agent focuses on forward reasoning. It allows Betty to hypothesize the outcome of various changes to the environment. For example, she may reason that if the number of plants in the river increases, then the amount of dissolved oxygen will increase. In the game environment, Betty needs to reason from given symptoms and problems, and hypothesize possible causes. To achieve this, the reasoning mechanism had to be extended to allow Betty to reason backward in the concept map structure. The combination of the forward and backward reasoner defines a diagnosis process [16] that was added to Betty's decision maker. The diagnosis component also gives Betty the capability of choosing the most probable cause when there are multiple possibilities of what is causing the problem in the river. Betty and the student can reflect on this information to decide on what additional information they need to determine the true cause for the problem they are working on.

4. Discussion and Future Work

In this paper, we have designed a game environment that combines the entertainment and flow provided by present day video games with innovative learning environments that support deep understanding of domain concepts, the ability to work with complex problems, and also develop metacognitive strategies that apply across domains. The Neverwinter Nights game interface and game engine are combined with the river ecosystem simulation to create a river adventure, where students solve a series of river ecosystem problems as they travel down a river. The learning by teaching component is retained, and incorporated into the game story by creating an initial phase where the student learns domain concepts and teaches Betty in a training academy. Components of the river adventure have been successfully tested, and preliminary experiments are being run on the integrated system. Our goal is to complete the preliminary studies this summer, and run a big study in a middle school classroom in Fall 2005.

Acknowledgements: This project is supported by NSF REC grant # 0231771.

References

[1] Provenzo, E.F. (1992). What do video games teach? *Education Digest*, 58(4), 56-58

[2] Lin, S. & Lepper, M.R. (1987). Correlates of children's usage of video games and computers. *Journal of Applied Social Psychology*, 17, 72-93.

[3] Squire, K. (2003). Video Games in Education. *International Journal of Intelligent Simulations and Gaming*, vol. 2, 49-62.

[4] Csikszentmihalyi, M. (1990). *Flow: The Psychology of Optical Experience*. New York: Harper Perrennial.

[5] Jonassen, D.H. (1988). Voices from the combat zone: Game grrlz talk back. In Cassell, J. & Jenkins, (Ed.), *From Barbie to Mortal Combat: Gender and Computer Games*. Cambridge, MA: MIT Press.

[6] Elliot, J., Adams, L., & Bruckman, A. (2002). No Magic Bullet: 3D Video Games in Education. *Proceedings of ICLS 2002*, Seattle, WA.

[7] Biswas, G., Schwartz, D., Bransford, J., & The Teachable Agents Group at Vanderbilt University. (2001). Technology Support for Complex Problem Solving: From SAD Environments to AI. In Forbus & Feltovich (eds.), *Smart Machines in Education*. Menlo Park, CA: AAAI Press, 71-98.

[8] Biswas, G., Leelawong, K., Belynne, K., et al. (2004). Incorporating Self Regulated Learning Techniques into Learning by Teaching Environments. in *The 26th Annual Meeting of the Cognitive Science Society*, (Chicago, Illinois), 120-125.

[9] Laird, J. & van Lent, M. The Role of AI in Computer Game Genres. http://ai.eecs.umich.edu/people/laird/papers/book-chapter.htm

[10] Leelawong, K., Wang, Y, Biswas, G., Vye, N., Bransford, J., & Schwartz, D. (2001). Qualitative reasoning techniques to support learning by teaching: The teachable agents project. *Proceedings of the Fifteenth International Workshop on Qualitative Reasoning* , San Antonio 73-80.

[11] Gupta, R., Wu, Y., & Biswas, G. (2005). Teaching About Dynamic Processes: A Teachable Agents Approach, Intl. Conf. on AI in Education, Amsterdam, The Netherlands, in review.

[12] Schwartz, D. & Martin, T. (2004). Inventing to Prepare for Future Learning: The Hidden Efficiency of Encouraging Original Student Production in Statistics Instruction. *Cognition and Instruction*. Vol. 22 (2), 129-184.

[13] White, B., Shimoda, T., Frederiksen, J. (1999). Enabling Students to Construct Theories of Collaborative Inquiry and Reflective Learning: Computer Support for Metacognitive Development. *International Journal of Artificial Intelligence in Education*, vol. 10, 151-182.

[14] BioWare Corp. (2002). *Neverwinter Nights* and *BioWare Aurora Engine*.

[15] Stieger Hardware and Softwareentwicklung. (2005). NeverwinterNights Extender 2

[16] Mosterman, P. & Biswas, G. (1999). Diagnosis of Continuous Valued Systems in Transient Operating Regions. *IEEE Transactions On Systems, Man, And Cybernetics—Part A: Systems And Humans*, Vol. 29(6),554-565.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Paper Annotation with Learner Models

Tiffany Y. Tang[1,2] and Gordon McCalla[2]
[1]*Dept. of Computing, Hong Kong Polytechnic University, Hong Kong*
cstiffany@comp.polyu.edu.hk
[2] *Dept. of Computer Science, University of Saskatchewan, Canada*
{yat751, mccalla}@cs.usask.ca

Abstract. In this paper, we study some learner modelling issues underlying the construction of an e-learning system that recommends research papers to graduate students wanting to learn a new research area. In particular, we are interested in learner-centric and paper-centric attributes that can be extracted from learner profiles and learner ratings of papers and then used to inform the recommender system. We have carried out a study of students in a large graduate course in software engineering, looking for patterns in such "pedagogical attributes". Using mean-variance and correlation analysis of the data collected in the study, four types of attributes have been found that could be usefully annotated to a paper. This is one step towards the ultimate goal of annotating learning content with full instances of learner models that can then be mined for various pedagogical purposes.

1. Introduction

When readers make annotations while reading documents, multiple purposes can be served: supporting information sharing [1], facilitating online discussions [2], encouraging critical thinking and learning [3], and supporting collaborative interpretation [4]. Annotations can be regarded as notes or highlights attached by the reader(s) to the article, and since they are either privately used or publicly shared by humans, and should thus ideally be in human-understandable format.

Another line of research on annotations focuses more on the properties (metadata) of the document as attached by editors (such as teachers or tutors in an e-learning context), e.g. using the Dublin Core metadata. Common metadata include Title, Creator, Subject, Publisher, References, etc. [5]. These metadata (sometimes referred to as item-level annotations) are mainly used to facilitate information retrieval and interoperability of the distributed databases, and hence need only be in machine-understandable format. Some researchers have studied automatic metadata extraction, where parsing and machine learning techniques are adapted to automatically extract and classify information from an article [6, 7]. Others have also utilized the metadata for recommending a research paper [8], or providing its detailed bibliographic information to the user, e.g. in *ACM DL* or *CiteSeer* [7]. Since those metadata are not designed for pedagogical purposes, sometimes they are not informative enough to help a teacher in selecting learning materials [9].

Our domain in this paper is automated paper recommendation in an e-learning context, with the focus on recommending technical articles or research papers with pedagogical value to learners such as students who are trying to learn a research area. In [10], we studied several filtering techniques and utilized artificial learners in recommending a paper to human learners. In that study, papers were annotated manually. The annotations included the covered topics, relative difficulty to a specific group of learners (senior undergraduate students), value-added (the amount of information that can be transferred to a student), and the authoritative level of the paper (e.g. whether the paper is well-known in the relevant area). The empirical results showed that learners' overall rating of a paper is affected by the helpfulness of the paper in achieving their goal, the topics covered by the paper, and the amount of knowledge gained after reading it. The study indicated that it is useful for a paper to be annotated by pedagogical attributes, such as what kinds of learners will like/dislike the paper or what aspects of the paper are useful for a group of learners. In this paper, we will describe a more extensive empirical analysis in pursuing an effective paper annotation for pedagogical recommendations.

In section 2, we will briefly describe the issues related to pedagogical paper recommendation and paper annotation; more information can be found in [10]. In section 3, we will describe the data used in our analysis. And in section 4, we will provide and discuss the results of our analysis. We make suggestions for further research in section 5.

2. Making Pedagogically-Oriented Paper Recommendations

A paper recommendation system for learners differs from other recommendation systems in at least three ways. The first is that in an e-learning context, there is a course curriculum that helps to inform the system. Since pure collaborative filtering may not be appropriate because it needs a large number of ratings (sparsity issue), the availability of a curriculum allows the deployment of a hybrid technique, partly relying on curriculum-based paper annotations. In addition, instead of relying on user feedbacks, we can also keep track of actual learner interactions with system to obtain implicit user models [11].

The second difference is the pedagogical issue. Beyond the learner interests, there are multiple dimensions of learner characteristics that should be considered in recommending learning material. For example, if a learner states that his/her interest is in *Internet Computing*, then recommending only the highly cited/rated papers in this area is not sufficient, because the learner may not be able to understand such papers. Thus, the annotations must include a wider range of learner characteristics.

The third difference comes from the rapid growth in the number of papers published in an area. New and interesting papers related to a course are published every year, which makes it almost impossible for a tutor to read all the papers and find the most suitable one for his/her learners. A bias in the annotations may also be generated if the paper is explicitly annotated by a teacher or tutor. Hence, an automated annotation technique is desirable. The benefit is not only to avoid bias through use of ratings by many readers, but also to reduce the workload of the human tutor.

For the purpose of automatic annotation, the source of information could come from either the content of the paper itself (intrinsic properties) or from the usage of the paper (extrinsic properties) by the readers. Usually, the intrinsic properties can be determined by using text processing or text mining techniques, e.g. the topics or subjects discussed in the paper, the difficulty level of the paper, or its authoritative level. But the extrinsic properties cannot be determined so readily, e.g. whether the paper is useful to learners, or contains value-added relative to any learner's knowledge.

In this paper, we will not focus on harvesting metadata of intrinsic properties from an existing paper library. Rather, we will focus on studying the collection of both intrinsic and extrinsic properties from learner experiences and feedback. What we are seeking are the pedagogical attributes that cannot be recognized easily. We argue here that relying on explicit metadata added to a digital library is not enough for the following reasons:

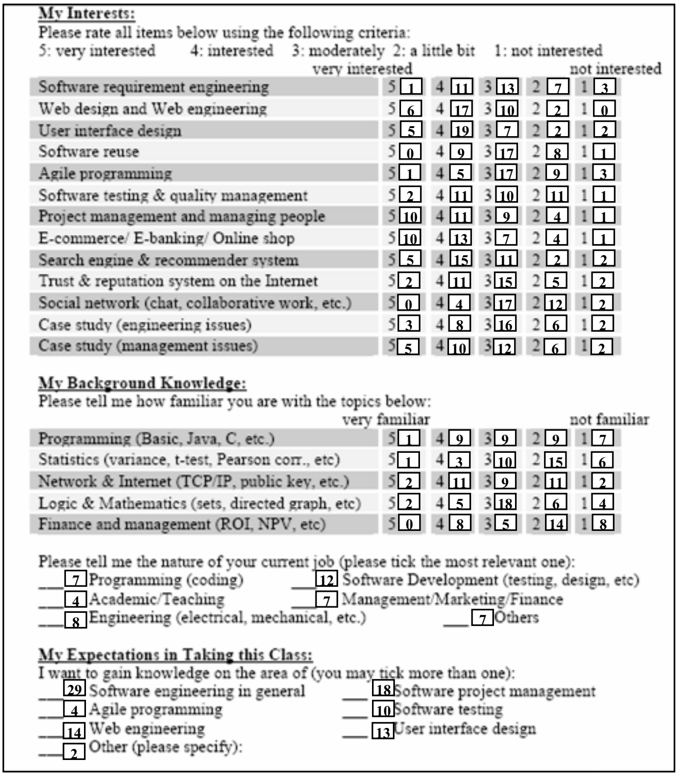
- The authoritative level of a paper is commonly determined by the number of citations of the paper or by the journal in which the paper is published. However, these are measures most useful for experienced researchers, whereas value to learners is determined by more diverse factors.

- Most learners have difficulty in specifying their interests, because they only have a superficial knowledge about the topics and may gain or lose interest in a topic after reading relevant or irrelevant papers. Additionally, the keywords or subjects provided by the metadata in a digital library usually represent a coarser-grained description of the topics, which may not match the details of a learner's interests.

In the next section we will describe a study in which papers were annotated with pedagogical attributes extracted from learner feedback and learner profiles, to see if learner-centered patterns of paper use can be found. This is another step in a research program aimed at annotating research papers with learner models, and mining these models to allow intelligent recommendations of these papers to students.

3. Data Collection

The study was carried out with students enrolled in a masters program in Information Technology at the Hong Kong Polytechnic University. In total 40 part-time students were registered in a course on Software Engineering (SE) in the fall of 2004, with curriculum designed primarily for mature students with various backgrounds. During the class, 22 papers were selected and assigned to students as reading assignments for 9 consecutive weeks starting from the 3rd until the 11th week. After reading them, students were required to hand in a feedback form along with their comments for each paper. In the middle of the semester, students were also asked to voluntarily to fill in a questionnaire (see Figure 1). 35 students returned the questionnaire and their data are analyzed here.



My Interests:
Please rate all items below using the following criteria:
5: very interested 4: interested 3: moderately 2: a little bit 1: not interested

	very interested				not interested
Software requirement engineering	5 1	4 11	3 13	2 7	1 3
Web design and Web engineering	5 6	4 17	3 10	2 2	1 0
User interface design	5 5	4 19	3 7	2 2	1 2
Software reuse	5 0	4 9	3 17	2 8	1 1
Agile programming	5 1	4 5	3 17	2 9	1 3
Software testing & quality management	5 2	4 11	3 10	2 11	1 1
Project management and managing people	5 10	4 11	3 9	2 4	1 1
E-commerce/ E-banking/ Online shop	5 10	4 13	3 7	2 4	1 1
Search engine & recommender system	5 5	4 15	3 11	2 2	1 2
Trust & reputation system on the Internet	5 2	4 11	3 15	2 5	1 2
Social network (chat, collaborative work, etc.)	5 0	4 4	3 17	2 12	1 2
Case study (engineering issues)	5 3	4 8	3 16	2 6	1 2
Case study (management issues)	5 5	4 10	3 12	2 6	1 2

My Background Knowledge:
Please tell me how familiar you are with the topics below:

	very familiar				not familiar
Programming (Basic, Java, C, etc.)	5 1	4 9	3 9	2 9	1 7
Statistics (variance, t-test, Pearson corr., etc)	5 1	4 3	3 10	2 15	1 6
Network & Internet (TCP/IP, public key, etc.)	5 2	4 11	3 9	2 11	1 2
Logic & Mathematics (sets, directed graph, etc)	5 2	4 5	3 18	2 6	1 4
Finance and management (ROI, NPV, etc)	5 0	4 8	3 5	2 14	1 8

Please tell me the nature of your current job (please tick the most relevant one):
___7 Programming (coding) ___12 Software Development (testing, design, etc)
___4 Academic/Teaching ___7 Management/Marketing/Finance
___8 Engineering (electrical, mechanical, etc.) ___7 Others

My Expectations in Taking this Class:
I want to gain knowledge on the area of (you may tick more than one):
___29 Software engineering in general ___18 Software project management
___4 Agile programming ___10 Software testing
___14 Web engineering ___13 User interface design
___2 Other (please specify):

Figure 1. Questionnaire for obtaining learner profile

3.1 Learners

Figure 1 shows the questionnaire and the frequencies of the answers by the students (the numbers inside the boxes on each question). The questionnaire has four basic categories: *interest, background knowledge, job nature,* and *learning expectation*. In each category we collected data about various features related to the subject of the course. We believe that these features constitute important dimensions of learners' pedagogical characteristics.

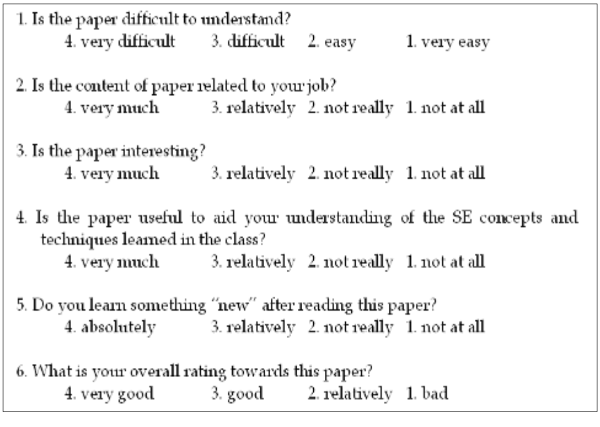
As shown in Figure 1, the population of learners has diverse interests, backgrounds, and expectations. As for their learning goals, most of the students expect to gain general knowledge about SE. But not all of them are familiar with programming (7 out of 35 say 'not familiar'). Hence, the students represent a pool of learners with working experience related to information technology, but do not necessarily have background in computer science.

3.2 Papers

The 22 papers given to the students were selected according to the curriculum of the course without considering the implications for our research (in fact, they were selected before the class began). All are mandatory reading materials for enhancing student knowledge. Table 1 tabulates the short description of some papers: the covered topics, the publication year, and the journal/magazine name of the publication.

Table 1. Short description of papers

Paper	**Topics**	**Year**	**Journal/magazine name**
#1	Requirements Eng.	2003	IEEE Software
#2	Project Mgmt.; Soft. Quality Mgmt.	2001	Comm. of the ACM
#3	Requirements Eng.	2003	IEEE Software
#6	Requirements Eng.; Agile Prog.; Project Mgmt.	2004	IEEE Software
#10	Web Eng.; UI Design	2001	IEEE Software
#11	Web Eng.; UI Design; Software Testing	2004	ACM CHI
#15	Web Eng.; UI Design; Soft. Testing; Case Study	1996	ACM CHI
#16	UI Design; SE in General	2003	ACM Interactions
#17	Web Eng.; Software Testing	1992	IEEE Computer
#20	Software Testing and Quality Mgmt.; Agile Prog.	2003	IEEE Software
#22	Project Mgmt.; Quality Mgmt.; Case Study	2004	IEEE Software



1. Is the paper difficult to understand?
   4. very difficult 3. difficult 2. easy 1. very easy

2. Is the content of paper related to your job?
   4. very much 3. relatively 2. not really 1. not at all

3. Is the paper interesting?
   4. very much 3. relatively 2. not really 1. not at all

4. Is the paper useful to aid your understanding of the SE concepts and techniques learned in the class?
   4. very much 3. relatively 2. not really 1. not at all

5. Do you learn something "new" after reading this paper?
   4. absolutely 3. relatively 2. not really 1. not at all

6. What is your overall rating towards this paper?
   4. very good 3. good 2. relatively 1. bad

Figure 2. Learner feedback form

3.3 Feedback

After reading each paper, students were asked to fill in a paper feedback form (Figure 2). Several features of the papers were to be evaluated by each student, including *its degree of difficulty to understand, its degree of job-relatedness with the user, its interestingness, its degree of usefulness, its ability to expand the user's knowledge (value-added),* and *its overall rating*. We used a Likert 4-scale rating for the answer.

4. Data Analysis and Discussion

Among the 35 students who answered the questionnaire, the vast majority read and rated all assigned papers. Table 2 shows the number who answered for each paper, along with the average overall ratings (Q.6 of Figure 2) and their standard deviations. From the table we can see that the range of average ratings is from 2.3 (paper #5) to 3.1 (paper #15), which means some papers are preferred over others, on average. Certainly, *the means and standard deviations of a paper's overall ratings* must be annotated to each paper and updated periodically, because this determines the general *quality* of a paper **(A1)**.

Table 2. Average overall ratings and number of observations

Paper	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**
Mean	2.8	2.9	2.4	2.5	2.3	2.9	3.0	2.8	3.0	2.9	2.6	2.8	2.7	2.9	3.1	2.4	3.0	2.8	2.9	2.6	2.8	2.9
StdD.	*.5*	*.6*	*.6*	*.7*	*.5*	*.8*	*.5*	*.5*	*.5*	*.5*	*.6*	*.5*	*.4*	*.8*	*.8*	*.6*	*.8*	*.6*	*.6*	*.7*	*.6*	*.5*
N	35	35	35	32	32	32	31	32	32	35	34	33	34	35	35	34	35	35	35	35	35	35

As shown in Table 1 some papers are on related topics, e.g. Web Engineering and UI design. Intuitively, if a learner likes/dislikes a paper on one topic, then s/he may like/dislike papers on similar topics. But this may not always be correct because the ratings may not depend exclusively on the topic of the paper. To check this, we have run a correlation analysis over the ratings of each pair of papers. The results show various correlations between -0.471 to 0.596 with 14 of them greater than or equal to 0.5 and only one less than -0.4. This suggests that some pairs of papers have moderately similar rating patterns, while others show an inverse pattern. The results can be used to generate recommendation rules across papers, such as:

- "If a learner likes paper #20 then s/he may like paper #21 with correlation 0.596"
- "If a learner likes paper #8 then s/he may dislike paper #13 with correlation 0.471"

Unsurprisingly, most high correlations are attained from the ratings of papers on different topics. If we pick the top-ten highest correlated ratings, only three pairs of papers belong to the same topics, i.e. (#14, #15), (#14, #17) and (#20, #21). Given this information, we propose to annotate a paper with *both positively and negatively correlated papers* **(A2)**.

To extract more information, a further analysis was performed by looking for patterns in student feedback on each paper, in particular looking for correlations between answers Q.1 to Q.5 on the feedback form (Figure 2) with Q.6 in order to determine the factors that affect a student's overall rating. Our conjecture is that the overall ratings given to each paper may uniquely be affected by those factors or a combination of them. For instance, some papers may get higher ratings due to having richer information about topics that match the interests of the majority of students, while others may get higher ratings due to good writing of the paper or its helpfulness to the student in understanding the concept being learned. If such patterns can be discovered, then we should be able to determine whether a particular paper is suitable to a particular learner based on the paper's and the learner's attributes. For instance, if the overall ratings of a paper have a strong correlation to learner interest, then we can recommend it to learners whose interests match the topic of the paper. Alternatively, if the ratings are strongly correlated to the learner's goal, then it will be recommended to learners with similar goals. Figure 3 illustrates the correlation of different factors, i.e. between Q.6 in Figure 2 with Q.1 to Q.5 for 22 papers. The Y-axis is the correlation coefficient with range [-1, 1].

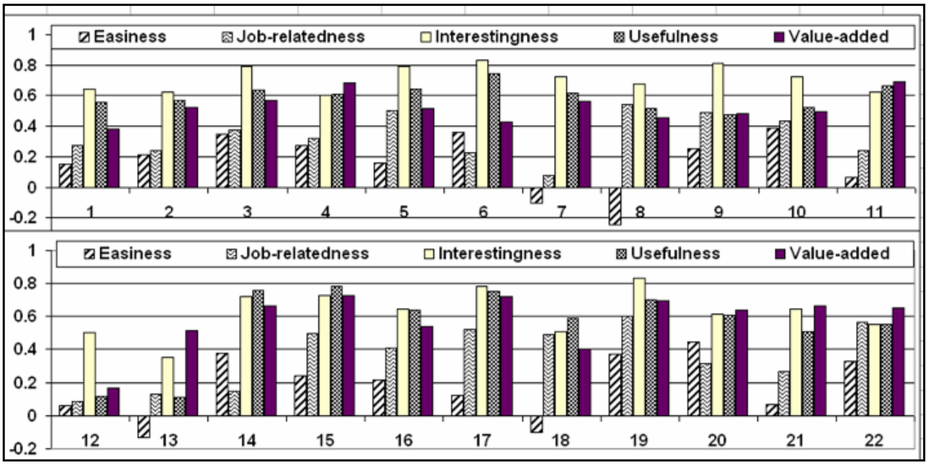
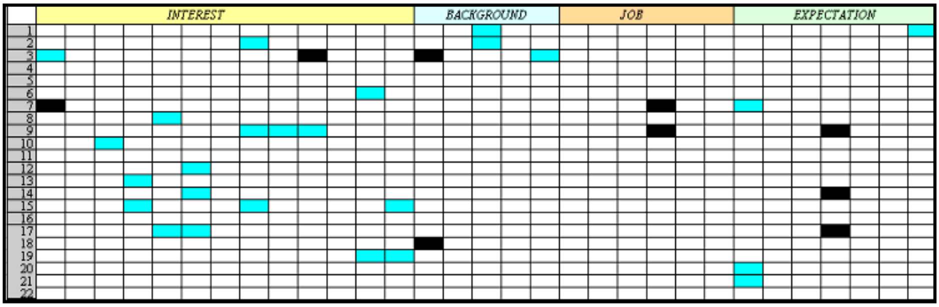


Figure 3. Factors that affect overall ratings

As shown in Figure 3, the learners' overall ratings of a paper are affected mostly by the interestingness of the paper, followed by the value-added gained after reading it and its usefulness in understanding the concept being learned. This result is slightly different from the result obtained in our prior study [10], where the usefulness slightly exceeded the interestingness. The reason is that in the current study we used a larger group of students and, more importantly, used different papers. As shown in Figure 3, the correlation varied for different papers, which means the individual differences of the papers matter here. Therefore, we also propose annotating a paper with the *correlation of the factors that affect learners' overall ratings* **(A3)**.

Finally, we can also determine the features of the learner (as determined by his or her questionnaire answers) that affect the learner's overall ratings. In other words, we analyze the correlations between the overall ratings and each feature in the learner's profile (Figure 1). Based on the conversion of the Likert scale, two methods are used simultaneously to extract the correlation. The first method is to convert the user interest and background knowledge into binary (3 to 5 into '1', and 1 and 2 into '0'), and assign '1' if the user ticks any feature in the 'job nature' and 'expectation' (see Figure 1). For the overall rating (Q.6 in Figure 2), '3' and '4' are interpreted as 'high' ratings; therefore we assign it a '1', while '1' and '2' are interpreted as 'low' ratings, and are assigned a '0'. After all values are converted into binary, then we run the correlation analysis. The second method is without converting any value to binary. We use both methods for the purpose of extracting more information.

Figure 4 shows the combined results of both methods. There are 22 rows, where each row represents a paper, and each column represents features of the learner profile shown in Figure 1 (taken top-down, e.g. 'job nature = software development' is the fourth column under JOB in Figure 4). If the correlation obtained from *either* method is greater than or equal to 0.4, the relevant cell is highlighted with a light color, and if it is smaller than or equal to -0.4, it is filled with a black color. If the correlation is in between (no significant correlation), then we have left it blank.



	INTEREST													BACKGROUND					JOB						EXPECTATION						
1																🟦															🟦
2								🟦								🟦															
3	🟦									⬛				⬛				🟦													
4																															
5																															
6												🟦																			
7	⬛																					⬛			🟦						
8					🟦																										
9								🟦	🟦	🟦												⬛						⬛			
10			🟦																												
11																															
12						🟦																									
13				🟦																											
14						🟦																						⬛			
15				🟦				🟦					🟦																		
16																															
17					🟦	🟦																						⬛			
18														⬛																	
19												🟦	🟦																		
20																									🟦						
21																									🟦						
22																															

Figure 4. The correlation matrix between overall ratings and learner models

It is shown from Figure 4 that only 16/22 papers have positive correlations with attributes of the learner profile. Some correlations can be verified easily, while others cannot. For instance, the ratings of the third paper are positively correlated to the first feature of learner interest (Q.1 in Figure 1: "software requirement engineering"). Yet the content of the third paper is about "requirements engineering" (cf. Table 1). And the ratings of the tenth paper (about "web engineering and UI design") are correlated to the third feature (about "UI design" too). Thus, by checking the positive correlation between learner ratings and their interests, we can infer the topics covered by the paper. However, this method also results in some unexplainable results, such as why there is a positive correlation between the ratings of paper #1 ("requirements engineering") with learners' expectations of learning UI design (the top-rightmost cell)? It also shows negative correlation between the ratings of paper #3 with learner interest in "trust and reputation system on the Internet", which cannot be explained even after checking the individual learner profiles. We think there are two possibilities here. The first is that the correlation is a coincidence, which may happen when the amount of data is small. The second is that the correlation represents hidden characteristics that have not been explained, something of interest discovered by the data mining. Due to limited data at the present time, we cannot derive any conclusion here. Nevertheless, we suggest annotating a paper with significant correlations of *the overall ratings with each feature of the learner profile* **(A4)**.

Given the pedagogical attributes (A1 – A4), we expect that the recommended papers can be more accurate and useful for learners. However, as in many recommendation systems, sparsity and scalability are two critical issues that may constrain a large-scale implementation. As the number of articles increases, the system may need to compute the correlations among thousands of documents, which in many cases cannot be completed real-time. Meanwhile, it is seldom that we can get enough learners to get a critical mass of ratings. Fortunately, both issues may not be so serious in e-learning systems. As pointed out earlier, the course curriculum may restrict the number of candidate papers within a subject and we can also utilize intrinsic properties to filter out irrelevant papers. In addition, low-rated and old papers will be discarded periodically, which eventually will increase the efficiency of the system.

Another concern comes from the reliability of the feedback, because learners may have their interests and knowledge changing over time. Intuitively, an extensive interaction between learners and system can be used to track these changing behaviours since many mandatory assessments are commonly used in any learning system. Instead of making an effort to solve this problem, we can trace these changes to provide us with a refined understanding about the usage of the paper and the learning curve of learners interacting with it.

5. Conclusions and Future Work

Several factors could affect the value of the annotations, including the properties of the paper and the learner characteristics. The combination of these properties then affects the learner ratings toward the paper. Through empirical analysis we have shown that we can use these correlations to extract paper properties by using the learner profiles and their paper ratings. Our data has also shown that the ratings of some papers have a significant correlation with the ratings of others and also attributes of learners.

So far, we have extracted four sets of pedagogical attributes (**A1** – **A4**) that can be annotated to a paper and used for recommendation. However, more information may still exist. For example, it may happen that the combinations of several learner attributes could better explain the learner ratings. In the future, we will use other data mining techniques to try to dig out such information, if it exists.

In the longer term this research supports the promise of annotating learning objects with data about learners and data extracted from learners' interactions with these learning objects. Such metadata may prove to be more useful, and perhaps easier to obtain, than metadata explicitly provided by a human tutor or teacher. This supports the arguments in [12] for essentially attaching instances of learner models to learning objects and mining these learner models to find patterns of end use for various purposes (e.g. recommending a learning object to a particular learner). This "ecological approach" allows a natural evolution of understanding of a learning object by an e-learning system and allows the e-learning system to use this understanding for a wide variety of learner-centered purposes.

Acknowledgements
We would like to thank the Canadian Natural Sciences and Engineering Research Council for their financial support for this research.

6. References

[1] Marshall, C. Annotation: from paper books to the digital library. *JCDL'97*, 1997.

[2] Cadiz, J., Gupta, A., and Grudin, J. Using web annotations for asynchronous collaborative around documents. *CSCW'00*, 2000, 309-318.

[3] Davis, J. and Huttenlocher, D. Shared annotation for cooperative learning. *CSCL'95*.

[4] Cox, D. and Greenberg, S. Supporting collaborative interpretation in distributed groupware. *CSCW'00*, 2000, 289-298.

[5] Weibel, S. The Dublin Core: a simple content description format for electronic resources. *NFAIS Newsletter*, 40(7):117-119, 1999.

[6] Han, H., Giles, C.L., Manavoglu, E. and Zha, H. Automatic document metadata extraction using support vector machines. *JCDL'03*, 2003, 37-48.

[7] Lawrence, S., Giles, C. L., and Bollacker, K. Digital libraries and autonomous citation indexing. *IEEE Computer*, 32(6): 67-71, 1999.

[8] Torres, R., McNee, S., Abel, M., Konstan, J.A. and Riedl, J. Enhancing digital libraries with TechLens. *JCDL'04*, 2004.

[9] Sumner, T., Khoo, M., Recker, M. and Marlino, M. Understanding educator perceptions of "quality" in digital libraries. *JCDL'03*, 2003, 269-279.

[10] Tang, T. Y., and McCalla, G.I. Utilizing artificial learners to help overcome the cold-start problem in a pedagogically-oriented paper recommendation system. *AH'04*, Amsterdam, 2004.

[11] Brooks, C., Winter, M., Greer, J. and McCalla, G.I. The massive user modeling system (MUMS). ITS'04, 635-645.

[12] McCalla, G.I. The ecological approach to the design of e-learning environments: purpose-based capture and use of information about learners. *J. of Interactive Media in Education (JIME)*, Special issue on the educational semantic web, T. Anderson and D. Whitelock (guest eds.), 1, 2004, 18p. [http://www-jime.open.ac.uk/2004/1]

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Automatic Textual Feedback for Guided Inquiry Learning

Steven TANIMOTO, Susan HUBBARD, and William WINN
Online Learning Environments Laboratory
Box 352350, Dept. of Computer Science and Engineering
University of Washington, Seattle, WA, 98195, USA

Abstract. We briefly introduce the online learning environment INFACT, and then we describe its textual feedback system. The system automatically provides written comments to students as they work through scripted activities related to image processing. The commenting takes place in the context of an online discussion group, to which students are posting answers to questions associated with the activities. Then we describe our experience using the system with a class of university freshmen and sophomores. Automatic feedback was compared with human feedback, and the results indicated that in spite of advantages in promptness and thoroughness of the automatically delivered comments, students preferred human feedback, because of its better match to their needs and the human's ability to suggest consulting another student who had just faced a similar problem.

1. Introduction

Timely feedback has been found in the past to improve learning [1]. However, it can be a challenge to provide such feedback in large classes or online environments where the ratio of users to teachers and administrators is high. We report here on an experimental system that provides automated feedback to students as they work on activities involving elementary image processing concepts.

1.1 Project on Intensive, Unobtrusive Assessment

The motivation for our project is to improve the quality of learning through better use of computer technology in teaching. We have focused on methods of assessment that use as their evidence not answers to multiple-choice tests but the more natural by-products of online learning such as students' user-interface event logs, newsgroup-like postings and transcripts of online dialogs. By using such evidence, students may spend more of their time engaged in the pursuit of objectives other than assessment ones: completing creative works such as computer programs and electronic art, or performing experiments using simulators in subject areas such as kinematics, chemical reactions, or electric circuits. (We currently support programming in Scheme and Python, and performing mathematical operations on digital images.)

Various artificial intelligence technologies have the potential to help us realise the goal of automatic, unobtrusive diagnostic educational assessment from evidence naturally available through online learning activities. These technologies include textual pattern matching, Bayesian inference, and Latent Semantic Indexing [4]. In this paper, we focus on our experience to date using textual pattern matching in this regard.

1.2 Facet-Based Pedagogy

Our project is studying automatic methods for educational assessment in a context in which multiple-choice tests are usually to be avoided. This means that other kinds of evidence must be available for analysis, and that such evidence must be sufficiently rich in information that useful diagnoses of learning impediments can be made. In order to obtain this quality of evidence, the learning activities in which our assessments are performed are structured according to a "facet-based pedagogy."

A *facet* is an aspect, conception, approximate state of understanding, or state of skill with regard to some concept, phenomenon, or skill. Minstrell [5] uses the term "facet" to refer to a variation of and elaboration of DiSessa's phenomenological primitive ("p-prim") [3]. We use the term "facet" in a more general sense, so as to be able to apply a general pedagogical approach to the learning not only of conceptual material such as Newton's laws of motion but also of languages and skills.

The facet-based pedagogical structure we use posits that instruction take place in units in which a cycle of teaching and learning steps proceeds. The cycle normally lasts one week. It begins with the posing of a problem (or several problems) by the instructor. Students then have one day to work on the problem individually and submit individual written analyses of the problem. Once these have been collected, students work in groups to compare and critique answers, keeping a record of their proceedings. By the end of the week, the students have to have submitted a group answer that incorporates the best of their ideas. It also must deal with any discrepancies among their individual analyses.

Students work in groups for several reasons. One is essentially social, allowing students to feel involved in a process of give-and-take and to help each other. Another is that the likely differences in students' thinking (assuming the problems are sufficiently challenging), will help them to broaden their perspectives on the issues and focus their attention on the most challenging or thought-provoking parts of the problem. And the most important reason, from the assessment point of view, to have the students work in groups is to help them communicate (to each other, primarily, as they see it, but also to us, indirectly) so as to create evidence of their cognition that we can analyze for misconceptions.

During the cycle, we expect some of the students' facets to change. The facets they have at the beginning of the unit, prior to the group discussion, are their preconceptions. Those they have at the end of the unit are their postconceptions. We want their postconceptions to be better than their preconceptions, and we want the postconceptions to be as expert-like as possible.

In order to facilitate teaching and learning with this facet-based pedagogy, we have developed a software system known as INFACT. We describe it in the next section.

2. The INFACT Online Learning Environment

Our system, called INFACT, stands for Integrated, Networked, Facet-based Assessment Capture Tool [6, 7]. INFACT catalyzes facet-based teaching and learning by (a) hosting online activities, (b) providing tools for defining specific facets and organising them, (c) providing simple tools for manual facet-oriented mark-up of text and sketches, (d) providing tools for displaying evidence in multiple contexts including threads of online discussion, and timeline sequence, and (e) providing facilities for automatic analysis and automatic feedback to students. INFACT also includes several class management facilities such as automatic assignment of student to groups based on the students' privately entered preferences (uses the Squeaky-Wheel algorithm), automatic account creation from class lists, and online standardized testing (for purposes such as comparison to the alternative means of assessment that we are exploring).

The primary source of evidence used by INFACT is a repository of evolving discussion threads called the *forum*. Most of the data in the forum is textual. However, sketches can be attached to textual postings, and user-interface log files for sessions with tools such as an image processing system known as PixelMath [8] are also linked to textual postings.

The forum serves the facet-based pedagogical cycle by mediating the instructor's challenge problem, collecting student's individual responses and hiding them until the posting deadline at which time the "curtain" is lifted and each student can see the posts of all members of his or her group. The forum hosts the ensuing group discussions, and provides a record of it for both the students and the instructor. Any facet-oriented mark-up of the students' messages made by the instructor or teaching assistants is also stored in the forum database. In the experiments we performed with manual and automated feedback to students, we used a combination of the forum and email for the feedback.

The facet-based pedagogy described above, as adapted for INFACT, is illustrated in Figure 1. A serious practical problem with this method of teaching is that the fourth box, "Teacher's facet diagnoses," is a bottleneck. When one teacher has to read all the discussions and interact with a majority of the students in a real class, most teachers find it impossible to keep up; there may be 25 or more students in a class, and teachers have other responsibilities than simply doing facet diagnoses. This strongly suggests that automation of this function be attempted.

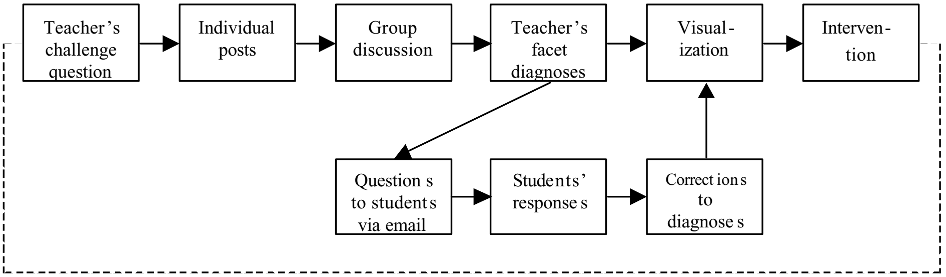
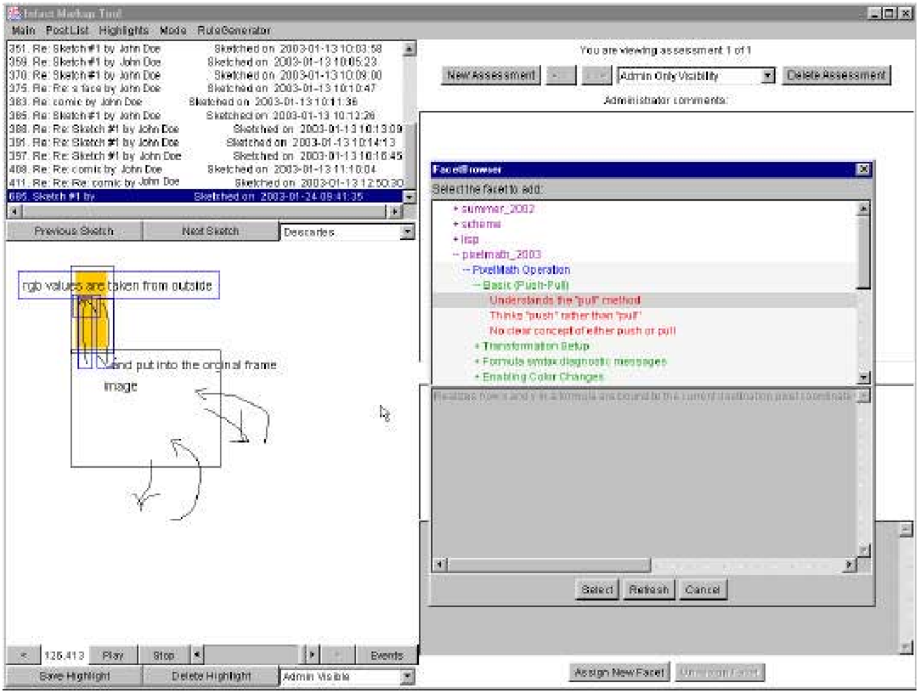


Figure 1. The INFACT pedagogical cycle. The period of the cycle is normally 1 week.

INFACT provides an interface for teachers to analyze student messages and student drawing, and create assessment records for the database and feedback for the students. Figure 2 illustrates this interface, selected for sketch-assessment mode. The teacher expresses an assessment for a piece of evidence by highlighting the most salient parts of the evidence for the diagnosis, and then selecting from the facet catalog the facet that best describes the student's apparent state of learning with regard to the current concept or capability.

In order to provide a user-customizable text-analysis facility for automatic diagnosis and feedback, we designed and implemented a software component that we call the INFACT rule system. It consists of a rule language, a rule editor, and a rule applier. The rule language is based



```
351. Re: Sketch #1 by John Doe          Sketched on 2003-01-13 10:03:58
359. Re: Sketch #1 by John Doe          Sketched on 2003-01-13 10:05:23
370. Re: Sketch #1 by John Doe          Sketched on 2003-01-13 10:09:00
375. Re: Re: a face by John Doe         Sketched on 2003-01-13 10:10:47
383. Re: comic by John Doe              Sketched on 2003-01-13 10:11:36
385. Re: Sketch #1 by John Doe          Sketched on 2003-01-13 10:12:26
388. Re: Re: Sketch #1 by John Doe      Sketched on 2003-01-13 10:13:09
391. Re: Re: Sketch #1 by John Doe      Sketched on 2003-01-13 10:14:13
397. Re: Re: Sketch #1 by John Doe      Sketched on 2003-01-13 10:16:45
408. Re: Re: comic by John Doe          Sketched on 2003-01-13 11:10:04
411. Re: Re: Re: comic by John Doe      Sketched on 2003-01-13 12:50:30
685. Sketch #1 by                       Sketched on 2003-01-24 09:41:35
```

You are viewing assessment 1 of 1

Administrator comments:

Select the facet to add:

- + summer_2002
- + scheme
- + lisp
- − pixelmath_2003
  - − PixelMath Operation
    - − Basic (Push-Pull)
      - Understands the "pull" method
      - Thinks "push" rather than "pull"
      - No clear concept of either push or pull
    - + Transformation Setup
    - + Formula syntax diagnostic messages
    - + Enabling Color Changes

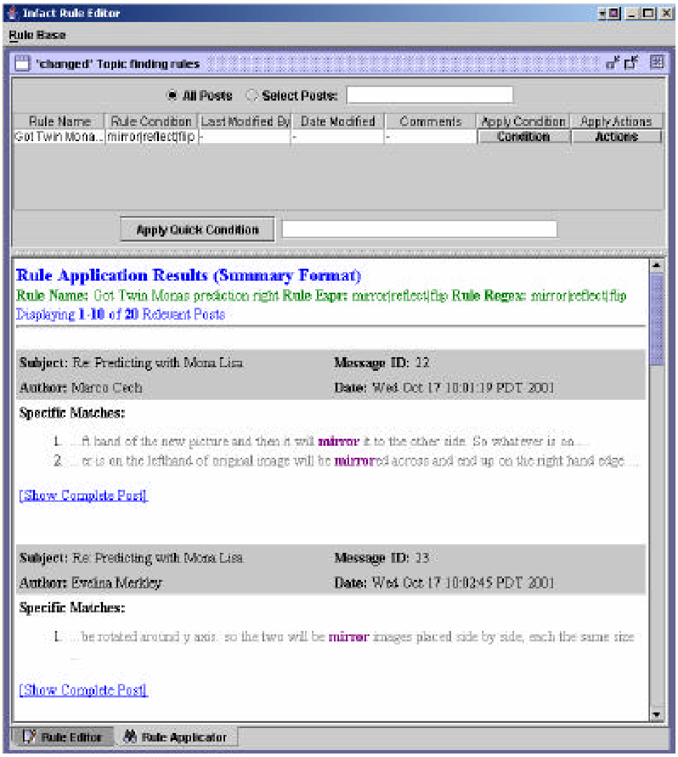
Realizes that x and y in a formula are bound to the current destination pixel coordinate

Figure 2. The manual mark-up tool for facet-based instruction. It is shown here in sketch-assessment mode, rather than text assessment mode.

on regular expressions with an additional construct to make it work in INFACT. The rule editor is a Java applet that helps assure that rules entered into the rule system are properly structured and written. The rule applier comprises a back-end Perl script and a Java graphical user interface.

The INFACT rule language is based on regular expressions. These regular expressions are applied by the rule applier to particular components of text messages stored in INFACT-Forum. In addition to the regular expressions, rule patterns contain "field specifiers." A field specifier identifies a particular component of a message: sender name, date and time, subject heading, body. Each instance of a field specifier will have its own regular expression. Someone creating a rule (e.g., a teacher or educational technology specialist) composes a rule pattern by creating any number of field specifier instances and supplying a regular expression for each one. Each field specifier instance and regular expression represent a subcondition for the rule, all of which must match for the rule to fire. It is allowed to have multiple instances of the same field specifier in a pattern. Therefore INFACT rules generalize standard regular expressions by allowing conjunction.

The rule applier can be controlled from a graphical user interface, and this is particularly useful when developing an assessment rule base. While regular expressions are a fundamental concept in computer science and are considered to be conceptually elementary, designing regular expressions to analyze text is a difficult and error-prone task, because of the complexity of natural language, particularly in the possibly broken forms typically used by students in online writing. Therefore we designed the rule applier to make it as easy as possible to test new rules. Although a complete rule specifies not only a condition, but also an action, the rule applier can be used in a way that safely tests conditions only. One can easily imagine that if it didn't have this facility, a teacher testing rules in a live forum might create confusion when the rules being debugged cause



Rule Name	Rule Condition	Last Modified By	Date Modified	Comments	Apply Condition	Apply Actions
Got Twin Mona...	mirror|reflect|flip	-	-	-	Condition	Actions

Rule Application Results (Summary Format)

Rule Name: Got Twin Monas prediction right **Rule Expr:** mirror|reflect|flip **Rule Regex:** mirror|reflect|flip

Displaying **1-10** of **20** Relevant Posts

Subject: Re: Predicting with Mona Lisa **Message ID:** 12

Author: Marco Cech **Date:** Wed Oct 17 10:01:19 PDT 2001

Specific Matches:

1. ...ft hand of the new picture and then it will **mirror** it to the other side. So whatever is on ...
2. ...er is on the lefthand of original image will be **mirror**ed across and end up on the right hand edge ...

[Show Complete Post]

Subject: Re: Predicting with Mona Lisa **Message ID:** 13

Author: Evelina Merkley **Date:** Wed Oct 17 10:02:45 PDT 2001

Specific Matches:

1. ...be rotated around y axis. so the two will be **mirror** images placed side by side, each the same size ...

[Show Complete Post]

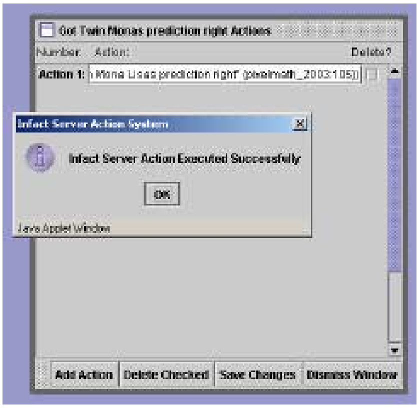
Figure 3. The “hit list” returned by the rule applier in testing mode.

email or INFACT postings to be sent to students inappropriately. When applying rules in this safe testing mode, the rule actions are not performed, and the results of condition matching are displayed in a “hit list” much like the results from search engine such as Google. This is illustrated in Figure 3. It is also possible to learn rules automatically [2], but this study did not use that facility.

3. The Study

The automated feedback system was tested in a freshman class for six weeks out of a ten-week quarter. The class was given in a small computer lab where each student had their own machine. Eighteen students completed the course and provided usable data. They were randomly divided into three groups, Arp, Botero and Calder. Almost all of the work discussed here was done collaboratively within these groups.

In addition to testing the usability and reliability of the automatic feedback system for instruction, the class was used to conduct a simple study in which the effectiveness of the automatic system was compared with the effectiveness of feedback provided by an instructor. A “no-feedback” condition served as a control. The three feedback conditions were rotated through the three groups using a within-subjects design so that every student had each kind of feedback for two weeks over the six-week period. The feedback study began with the fourth week of class. The order of the types of feedback was different for each group. Each two-week period required the students to complete exercises in class and as homework. Every week, activities were



Got Twin Monas prediction right Actions

Number: Action: Delete?

Action 1: n Mona Lisas prediction right" (pixelmath_2003:105)) □

Infact Server Action System

Infact Server Action Executed Successfully

Figure 4. Feedback to the teacher/administrator from the action subsystem of the rule system.

assigned requiring each student to find the solution to a problem set by the instructor (a PixelMath formula, a strategy, some lines of Scheme code) and to post that solution to INFACT-Forum by mid-week. The group then had the rest of the week to come to a consensus on the solution and to post it. At the end of the two-weeks, before the groups rotated to the next type of feedback, students took a short on-line post-test over the content covered in the preceding two weeks.

The automatic feedback was provided in the manner described above. The human feedback was provided by an instructor ("Alan"). During the class, Alan sat at one of the lab computers watching posts come into INFACT-Forum from the group whose turn it was to receive human feedback. As each post arrived, he responded. Out of class, Alan checked the forum every day and responded to every post from the designated group. Students in the no-feedback group were left to their own devices.

Several data sources were available, including scores on the post-tests, the students' posts and the feedback provided automatically and by Alan, interviews with selected students at the end of each two-week period conducted by a research assistant, questionnaires, and observations of the class by three research assistants. The class instructor and Alan were also interviewed.

4. Findings

Analysis of the post-test scores showed no statistically reliable differences among the groups as a function of the type of feedback they received, nor significant interactions among group, feedback, or the order in which the groups received feedback. There are two explanations for this finding, aside from taking it as evidence that the automatically-provided feedback was neither more nor less effective than that provided by Alan, and that neither was better than no feedback. First, the small number of students in each group reduced the statistical power of the analysis to the point where type-two errors were a real possibility. Second, the first no-feedback group was quick to organize itself and to provide mutually-supporting feedback within its members. This proved to be extremely effective for this group (Arp) and subsequently also for Botero and Calder when it was their turn not to receive feedback.

However, examination of other data sources showed some differences between the automatic and Alan's feedback, as well as some similarities. First, both encountered technical problems. For the first few sessions, the automatic feedback system was not working properly.

This made it necessary for a research assistant to monitor the posts from the automatic feedback group and to decide from the rules which prepared feedback statement to send. Fortunately, the bug was fixed and the Wizard-of-Oz strategy was quickly set aside. Also, Alan soon discovered that posting his feedback to INFACT-Forum took too long as the system acted sluggishly. It was therefore decided to send the "human" feedback to the students' personal email accounts. This was much quicker. However, it required the students to have their email programs open at the same time as INFACT-Forum and PixelMath. With so many windows open, some students did not notice Alan's feedback until some time after it had been sent. Some even minimized their email windows to make their screens more manageable and did not read the feedback until some time after it was sent, if at all.

The most obvious difference between the automatic and the human feedback was that the automatic feedback was very quick, while it took Alan time to read students' posts, consider what to reply, to type i and send it. This delay caused some minor frustration. One observer reported students posting to INFACT and then waiting for Alan's response before doing anything else. Several students were even seen to turn in their seats and watch Alan from behind while they were waiting for feedback. Also, out of class, Alan's feedback was not immediate, as he only checked the forum once a day. Automatic feedback was provided whenever a student posted something, whether during class or out of class.

Next, the automatic feedback responses were longer and more detailed than Alan's. This was because they had been generated, with careful thought, ahead of time, while Alan responded on the fly. Alan also mentioned that he often had difficulty keeping up with the student posts during class and that he had to be brief in order to reply to them all.

Over the six weeks Alan posted close to 300 messages. The automatic system sent less than 200. The main reason for this difference seems to be Alan's tendency to respond in a manner that encouraged the development of discussion threads. While both types of feedback asked questions of students and asked them to post another message as a matter of course ("Why do you think that is?", "Try again and post your response."), this tactic produced only one follow-on post to an automatic feedback message during the six weeks of the study.

Though posting shorter messages, Alan was better than the automatic system at deciding what a student's particular difficulty might be, and responding more flexibly and particularly to individual students' posts. Some of the students said they preferred Alan's feedback for this reason, finding the automatic feedback too general or less directly relevant to their particular difficulties or successes. Moreover, Alan could sometimes determine more precisely than the automatic system what was causing a student to have a problem. In such cases, he would often suggest a strategy for the student to try, rather than giving direct feedback about the student's post. Alan also referred students to other students' posts as part of his feedback. Because he was monitoring all of the posts from the group, while the students themselves might not be, he knew if another student had solved a problem or come up with a suggestion that would be useful to the student to whom he was currently responding, and did not hesitate to have the student look at the other's post. This also speeded up the feedback process somewhat. On two occasions, Alan was able to spot common problems that were then addressed for everyone in the next class session.

The students found Alan's feedback more personal. He made typos and used incomplete sentences. The automatic system did not. He used more vernacular and his posts reflected a more friendly tone. Alan also made an occasional mistake in the information he provided through feedback, though, fortunately, these were quickly identified and put right. In spite of this, most students preferred interacting with a human rather than the automatic system.

Finally, as we mentioned above, the first group to receive no feedback, Arp, compensated for this by providing feedback and other support to each other. By coincidence, students in Arp, more than in Botero and Calder, had, by the fourth week, developed the habit of helping each other through the forum. It turns out that Arp also contained the strongest students in the class who, collectively, had strength in all the skills required in the course. As a result, requests for help from one group member were answered without fail, in one case by ten responses from the other group members. One result of this was that, when it was Arp's turn to receive the system's feedback and then Alan's, they had come to rely on it. (The students who stopped work until Alan replied to their posts, whom we mentioned above, were all from Arp.)

To summarize, the automatic feedback system delivered feedback and showed potential. Initial technical problems were quickly solved and the students received detailed and mostly relevant feedback on their posts to INFACT-Forum. The comparison to human feedback points to improvements that should be considered. First, it would be useful if the system could cross-reference student posts so that students could be referred to each other's contributions in a way that proved effective in Alan's feedback. More generally, the ability of feedback from the automatic system to generate more collaboration among the students would be an important improvement. Second, the ability of the system to better diagnose from posts the reasons students were having problems would be useful. This would allow the system to sustain inquiry learning for more “turns” in the forum, rather than giving the answer, or suggesting a particular strategy to try. Third, any changes that made the automatic system appear to be more human would make it better received by students. Finally, it would be nice to create a computer-assisted feedback system in which the best of automated and human faculties can complement one another.

Acknowledgments

The authors wish to thank E. Hunt, R. Adams, C. Atman, A. Carlson, A. Thissen, N. Benson, S. Batura. J. Husted, J. Larsson, D. Akers for their contributions to the project, the National Science Foundation for its support under grant EIA-0121345, and the referees for helpful comments.

References

[1] Black, P., and Wiliams, D. 2001. Inside the black box: Raising standards through classroom assessment. Kings College London Schl. of Educ. http://www.kcl.ac.uk/depsta/education/publications/Black%20Box.pdf.

[2] Carlson, A., and Tanimoto, S. 2003. Learning to identify student preconceptions from text, *Proc. HLT/NAACL 2003 Workshop: Building Educational Applications Using Natural Language Processing*.

[3] diSessa, A. 1993. Toward an epistemology of physics. *Cognition & Instruction*, 10, 2&3, pp.105-225.

[4] Graesser, A.C., Person, N., Harter, D., and The Tutoring Research Group. 2001a. Teaching tactics and dialog in AutoTutor. *International Journal of Artificial Intelligence in Education.*

[5] Minstrell, J. 1992. Facets of students' knowledge and relevant instruction. In Duit, R., Goldberg, F., and Niedderer, H. (eds.), *Research in Physics Learning: Theoretical Issues and Empirical Studies*. Kiel, Germany: Kiel University, Institute for Science Education.

[6] Tanimoto, S. L., Carlson, A., Hunt, E., Madigan, D., and Minstrell, J. 2000. Computer support for unobtrusive assessment of conceptual knowledge as evidenced by newsgroup postings. *Proc. ED-MEDIA 2000*, Montreal, Canada, June.

[7] Tanimoto, S., Carlson, A., Husted, J., Hunt, E., Larsson, J., Madigan, D., and Minstrell, J. 2002. Text forum features for small group discussions with facet-based pedagogy, *Proc. CSCL 2002*, Boulder, CO.

[8] Winn, W., and Tanimoto, S. 2003. On-going unobtrusive assessment of students learning in complex computer-supported environments. Presented at Amer. Educ. Res. Assoc. Annual Meeting, Chicago IL.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Graph of Microworlds: A Framework for Assisting Progressive Knowledge Acquisition in Simulation-based Learning Environments

Tomoya Horiguchi* Tsukasa Hirashima**
*Faculty of Maritime Sciences, Kobe University
**Deptartment of Information Engineering, Hiroshima University

Abstract: A framework for assisting a learner's progressive knowledge acquisition in simulation-based learning environments (SLEs) is proposed. In SLE, usually a learner is first given a simple situation to acquire basic knowledge, then given more complicated situation to refine it. Such change of situation often causes the change of the model to be used. Our GMW (graph of microworlds) framework effifiently assists a learner in such 'progressive' knowledge acquisition by adaptively giving her/him microworlds. A node of GMW has the description of a microworld which includes the model, its modeling assumptions (which can explain why the model is valid in the situation) and the tasks through which one can understand the model. The GMW, therefore, can adaptively provide a learner with the microworld and the relevant tasks to understanding it. An edge has the description of the difference/change between microworlds. The GMW, therefore, can provide the relevant tasks which encourage a learner to transfer to the next microworld and can explain how/why the behavioral change of the model is caused by the change of the situation in model-based way. This capability of GMW greatly helps a learner progressively refine, that is, reconstruct her/his knowledge in a concrete context.

1. Introduction

Simulation-based learning environments (SLEs) have a great potential for facilitating exploratory learning: a learner could act on various objects in the environment and acquire knowledge in a concrete manner. However, it is difficult for most learners to be engaged in such learning activities by themselves. The assistance is necessary at least by providing the relevant task and settings through which a learner encounters new facts and apply them. The task, in addition, should be always challenging and accomplishable for a learner. With this view, a popular way is to provide a series of increasingly complex tasks through the progression of learning. Typically, in SLEs, a learner is first provided with a simple example and some exercises similar to it to learn some specialized knowledge, then provided with more complex exercises to refine the knowledge. This 'genetic' [11] approach has been generally used in SLEs for designing instruction [13][16][17].

The exercises to learn the specialized knowledge in SLEs means the situations in which a learner has to consider only a few conditions about the phenomena. The exercises to refine the knowledge means the situations in which she/he has to consider many conditions. In other words, the models are different which are necessary to think about the phenomena in SLEs. Therefore, it is reasonable to segment the domain knowledge into multiple models of different complexity, which is the basic idea of 'ICM (increasingly complex microworlds)' approach [3][7]. In ICM, a learner is introduced to a series of increasingly complex microworlds step by step, each of which has the simplified/focused domain model to its degree. This makes it easier to prevent a learner from encountering too difficult situations during exploration and to isolate the error about a segment of knowledge from the others, which greatly helps debug a learner's misunderstandings. Several systems have been developed according to ICM approach and their usefulness has been verified [7][18][19][20][21].

The limitations of these systems are that they have little adaptability, and that they can hardly explain the differences between the models. It is important to adaptively change the situation to each learner's knowledge state, her/his preference, the learning context etc. It is also important to explain why the new or more refined knowledge is necessary in the new situation. Though the existing ICM-based systems are carefully designed for progressive knowledge acquisition, the target knowledge of each microworld and the tasks for acquiring it isn't necessarily explicitly represented on the system (The target knowledge of a microworld means its model. We say 'a learner has understood the model' in the same meaning as 'she/he has acquired the target knowledge'). This makes it difficult to customize the series of microworlds for each learner, and to explain the necessity of microworld-transitions. In order to address these problems, the followings have to be explicitly represented: (1) the target knowledge of each microworld and the tasks for acquiring it, and (2) the difference of the target knowledge between the microworlds and the tasks for understanding it.

In this paper, we propose a framework for describing such target knowledge and tasks of a series of microworlds to assist progressive knowledge acquisition. It is called 'graph of microworlds (GMW)': the graph structure the nodes of which stand for the knowledge about microworlds and the edges of which stand for the knowledge of the relation between them.

By using the item (1), the GMW-based system can identify the microworlds for a learner to work on next, and provide the relevant tasks for her/him to acquire the target knowledge in each microworld. By using the item (2) (especially because it is described in model-based way), the system can provide the relevant tasks for encouraging a learner to transfer to the next microworld, and explain the necessity of the transition in model-based way. For example, the task is provided in which the previous model isn't applicable but the new or more refined model is necessary. If a learner made a wrong solution by using the previous model, the system explains why her/his solution is wrong by relating it to the difference between the previous and new models, that is, the difference of models in two microworlds. This capability of the system would greatly help a learner progressively reconstruct her/his knowledge in a concrete context.

In fact, there have been developed several SLEs which have multiple domain models. Such systems embody the ICM principle to some extent whether they refer to it or not. In QUEST [21], ThinkerTools [18][19][20] and DiBi [14], for example, a series of microworlds are designed to provide a learner with increasingly complex situations and tasks which help her/him acquire the domain knowledge progressively (e.g., from qualitative to quantitative behavior, from voltage value to its change, from uniform (frictionless) to decelerated (with friction) motion). In 'intermediate model' [9][10] and WHY [5][15], on the other hand, a set of models are designed from multiple viewpoints to explain the knowledge of a model by the one of another model which is easier to understand (e.g., to explain the macroscopic model's behavior as the emergence from its microscopic model's one).

These systems, however, have the limitations described above. They usually have only a fixedly ordered series of microworlds. If one would use them adaptively, human instructors are necessary who can determine which microworld a learner should learn next and when she/he should transfer to it. Even though it is possible to describe a set of rules for adaptively choosing the microworlds, the rules which aren't based on the differences of models couldn't explain the 'intrinsic' necessity of transition. This is also the case about the recent non-ICM-based SLEs with sophisticatedly designed instruction [13][16][17]. Their frame-based way of organizing the domain and instructional knowledge often makes the change of tasks or situations in instruction 'extrinsic.'

The GMW framework addresses these problems by explicitly representing the knowledge about the microworlds and the difference between them in terms of their models, situations, viewpoints, applicable knowledge and the tasks for acquiring it.

2. GMW: The Graph of Microworlds

2.1 Specification for the Description of Microworlds

In microworlds, a learner is required not only (t1) to predict the behavior of the physical system in a situation, but also (t2) to predict the change of behavior of the system given the change of the situation. That is, there are two types of tasks each of which requires (t1) and (t2) respectively. The latter is essential for a learner to refine her/his knowledge because the change of the situation might change the model itself to be used for prediction. A learner should get able not only to accomplish the task by using a model, but also to do so by choosing the relevant model to the given situation. Our research goal is, therefore, (1) to propose a framework for describing a set of models and the differences/changes between them and, based on this description, (2) to design the functions which adaptively provide a learner with microworlds (i.e., situations and tasks) and explain how/why the models change according to the changes of situations.

The model of a physical system changes when the situation goes out of the range within which it is valid. The range can be described as the modeling assumptions, which are the assumptions necessary for the model to be valid. In this research, we consider the followings[*1]:

(a1) the physical objects and processes considered in a model
(a2) the physical situation of the system (e.g., a constraint on the parameters' domains/values, the structural conditions of the system)
(a3) the behavioral range of the system to be considered (e.g., the interval between boundary conditions, the mode of operation)
(a4) the viewpoint for modeling the system (e.g., qualitative/quantitative, static/dynamic)

The change of modeling assumptions causes the model of physical system to change. From the educational viewpoint, it is important to causally understand a behavioral change of physical system related to its corresponding change of modeling assumptions. Therefore, our framework should include not only the description of (the change of) models but also the description of (the change of) modeling assumptions. In addition, it should also include the description of the tasks which trigger the change of models, that is, encourage a learner to think about the differences of models.

Based on the discussion above, we propose the framework for describing and organizing microworlds in section 2.2.

2.2 Description and Organization of Microworlds

2.2.1 Description of a Microworld

The following information is described in each microworld.

(m1) the target physical system and a model of it.
(m2) the physical objects and processes to be considered in the model (a1)
(m3) the physical situation of the system (a2)
(m4) the behavioral range of the system (a3) and the viewpoint for the modeling (a4)
(m5) the skills necessary for the model-based inference
(m6) the tasks and the knowledge necessary for accomplishing them.

The items (m2), (m3) and (m4) stand for the valid combination of modeling assumptions which corresponds to a (valid) model of the physical system (m1). The item (m5) stands for the skills used with the model for accomplishing tasks (e.g., numerical calculation for a quantitative model). The item (m6) stands for the tasks to be provided for a learner, to each of which the knowledge necessary for accomplishing it (the subset of

[*1] We reclassified the modeling assumptions discussed in [6].

(m1)-(m5)) is attached.

From the viewpoint of model-based inference, there are two types of tasks: the task which can be accomplished by using the model of the microworld it belongs to, and the task which needs the transition to another microworld (that is, which needs another model) to be accomplished. All of the task (t1) are the former type. The tasks (t2) which don't need the change of the model (i.e., the given change of conditions doesn't cause the change of modeling assumptions) are also the former type. They are called 'intra-mw-tasks.' The knowledge necessary for accomplishing an intra-mw-task can be described by using (m1)-(m5) of the microworld it belongs to. The tasks (t2) which need the change of the model (i.e., the given change of conditions causes the change of modeling assumptions) are the latter type. They are called 'inter-mw-tasks.' The knowledge necessary for accomplishing an inter-mw-task is described by using (m1)-(m5) of the microworld it belongs to and (m1)-(m5) of the microworld to be transferred to. The description of inter-mw-task includes the pointer to the microworld to be transferred to.

2.2.2 Organization of Microworlds

In order to organize the set of microworlds as described above, we propose the 'Graph of Microworlds (GMW).' The GMW makes it possible to adaptively generate the series of microworlds to each learner. It is the extension of the 'Graph of Models (GoM)' [1][2] which is the framework for describing how the model of a physical system can change by the change of its constraints. The nodes of GoM stand for the possible models of the system and its edges stand for the changes of modeling assumptions (which are called 'model-transitions'). The GoM is applied to model identification by observational data, fault diagnosis etc.

We extend the GoM to be the GMW the nodes of which stand for the microworlds and the edges of which stand for the possible transitions between them. Two educational concepts are introduced into GMW: the knowledge which a learner could acquire by understanding the model of a microworld, and the task by accomplishing which she/he could understand the model. The target knowledge of a microworld is its model, modeling assumptions and the skills used with the model (i.e., (m1)-(m5)). In order to encourage a learner to acquire it, the system provides her/him with the intra-mw-tasks of the microworld.

In order to encourage a learner to transfer to another microworld, on the other hand, the system provides her/him with the inter-mw-task, the target knowledge of which is the difference between the knowledge about the two models. In GMW, two nodes have the edge between them if the difference between their target knowledge is sufficiently small (i.e., the transition between two microworlds is possible if it is educationally meaningful as the evolution of models). In the neighborhood of a microworld, therefore, there are a few microworlds which are similar to it in terms of the target knowledge. This makes it possible for the system to adaptively choose the next microworld according to the learning context.

(Example-1) Curling-like Problem (1)

Figure 1a shows a 'curling-like' situation. At the position x_0, a stone M_1 is thrown by a player with the initial velocity v_0, then slides on the ice rightward until it collides with another stone M_2 at the position x_1. If the friction on the ice isn't negligible and the initial velocity is small, it may stop between x_0 and x_1 (described as 'the interval $[x_0, x_1]$') without collision. By the player's decision, the interval $[x_0, x_1]$ may be swept with brooms (uniformly) before the start of M_1.

When modeling the behavior of this physical system, there can be various physical situations (e.g., the initial velocity is small/large, the friction is/isn't negligible, the ice is/isn't swept), behavioral ranges (e.g., the interval before/after the collision, the instant of collision) and viewpoints (e.g., qualitative/quantitative). Therefore, several models are constructed corresponding to them. These models are, with the tasks for understanding them, then organized into the GMW (as shown in Figure 1b). Some of the modeling assumptions and tasks in the microworlds are described as follows:

MW-1: *(m1)* $v_1(t) = v_0$, $x_1(t) = x_0 + v_0 t$
(m2) uniform motion (no force works on M_1)
(m3) $0 < v_0 < v_0^1$, $\mu_1 <$ *epsilon, not sweep*($[x_0, x_1]$)
(m4) position(M_1) *is in* $[x_0, x_1]$
(m5) numerical calculation
(m6) (1) derive the velocity of M_1 at the position x ($x_0 < x < x_1$).
(2) derive the velocity of M_1 at the position x ($x_0 < x < x_1$) when it becomes $\mu_1 >$ epsilon. [-> MW-2:(m6)-(1)]*
(3) derive the velocity of M_1 after the collision with M_2 when it becomes $v_0 > v_0^1$ (assume the coefficient of restitution e = 1). [-> MW-4:(m6)-(1)]*

MW-2: *(m1)* $a_1(t) = -\mu_1 M_1 g$, $v_1(t) = v_0 - \mu_1 M_1 gt$, $x_1(t) = x_0 + v_0 t - \mu_1 M_1 gt^2/2$
(m2) uniformly decelerated motion, frictional force from the ice
(m3) $0 < v_0 < v_0^1$, $\mu_1 >$ *epsilon, not sweep*($[x_0, x_1]$)
(m4) position(M_1) *is in* $[x_0, x_1]$
(m5) numerical calculation
(m6) (1) derive the velocity of M_1 at the position x ($x_0 < x < x_1$).
(2) derive the position x ($x_0 < x < x_1$) at which M_1 stops.
(3) derive the position x ($x_0 < x < x_1$) at which M_1 stops when the interval $[x_0, x_1]$ is swept. [-> MW-3:(m6)-(1)]*
(4) derive the velocity of M_1 after the collision with M_2 when it becomes $v_0 > v_0^1$ (assume the coefficient of restitution e = 1). [-> MW-4:(m6)-(1)]*

MW-3: *(m1)* $a_1(t) = -\mu_2 M_1 g$, $v_1(t) = v_0 - \mu_2 M_1 gt$, $x_1(t) = x_0 + v_0 t - \mu_2 M_1 gt^2/2$
(m2) uniformly decelerated motion, frictional force from the ice, heat generation by sweeping, melt of the surface of the ice by the heat (which makes the coefficient of friction decrease to μ_2 and the temperature of the surface of ice increase to zero centigrade degree)

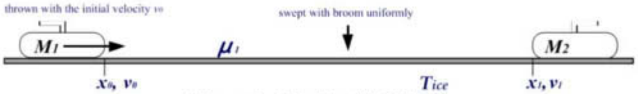
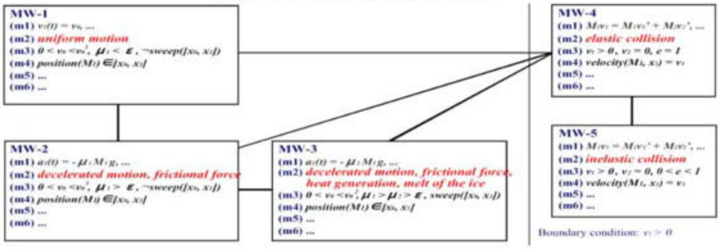

thrown with the initial velocity v_0
swept with broom uniformly
M_1
μ_1
M_2
x_0, v_0
T_{ice}
x_1, v_1

Figure 1a. ‘Curling-like’ Situation


MW-1
(m1) $v_1(t) = v_0$, ...
(m2) uniform motion
(m3) $0 < v_0 < v_0^1$, $\mu_1 < \varepsilon$, ¬sweep([x_0, x_1])
(m4) position(M_1) ∈[x_0, x_1]
(m5) ...
(m6) ...
MW-2
(m1) $a_1(t) = -\mu_1 M_1 g$, ...
(m2) decelerated motion, frictional force
(m3) $0 < v_0 < v_0^1$, $\mu_1 > \varepsilon$, ¬sweep([x_0, x_1])
(m4) position(M_1) ∈[x_0, x_1]
(m5) ...
(m6) ...
MW-3
(m1) $a_1(t) = -\mu_2 M_1 g$, ...
(m2) decelerated motion, frictional force, heat generation, melt of the ice
(m3) $0 < v_0 < v_0^2$, $\mu_1 > \mu_2 > \varepsilon$, sweep([$x_0$, x_1])
(m4) position(M_1) ∈[x_0, x_1]
(m5) ...
(m6) ...
MW-4
(m1) $M_1v_1 = M_1v_1' + M_2v_2'$, ...
(m2) elastic collision
(m3) $v_1 > 0$, $v_2 = 0$, $e = 1$
(m4) velocity(M_1, x_1) = v_1
(m5) ...
(m6) ...
MW-5
(m1) $M_1v_1 = M_1v_1' + M_2v_2'$, ...
(m2) inelastic collision
(m3) $v_1 > 0$, $v_2 = 0$, $0 < e < 1$
(m4) velocity(M_1, x_1) = v_1
(m5) ...
(m6) ...
Boundary condition: $v_1 > 0$

(m3) $0 < v_0 < v_0^2$, $\mu_1 > \mu_2 >$ *epsilon, sweep*($[x_0, x_1]$)
(m4) *position*(M_1) *is in* $[x_0, x_1]$
(m5) *numerical calculation*
(m6) (1) *derive the position x* ($x_0 < x < x_1$) *at which* M_1 *stops.*

MW-4: (m1) $M_1v_1 = M_1v_1' + M_2v_2$, $-(v_1' - v_2')/(v_1 - v_2) = e$
(m2) *elastic collision, the total kinetic energy is conserved*
(m3) $v_1 > 0$, $v_2 = 0$, $e = 1$
(m4) *velocity*(M_1, x_1) $= v_1$
(m5) *numerical calculation*
(m6) (1) *derive the velocity of* M_1 *after the collision with* M_2.
(2*) *derive the velocity of* M_1 *after the collision with* M_2 *when it becomes* $0 < e < 1$. *[-> MW-5:(m6)-(1)]*

MW-5: (m1) $M_1v_1 = M_1v_1' + M_2v_2$, $-(v_1' - v_2')/(v_1 - v_2) = e$
(m2) *inelastic collision, deformation of the stones by collision (which makes the total kinetic energy decrease)*
(m3) $v_1 > 0$, $v_2 = 0$, $0 < e < 1$
(m4) *velocity*(M_1, x_1) $= v_1$
(m5) *numerical calculation*
(m6) (1) *derive the velocity of* M_1 *after the collision with* M_2.

where,

1. v_0^1 and v_0^2 are the minimal initial velocities of M_1 for the collision to occur when the coefficients of friction are μ_1 and μ_2 respectively.
2. If the coefficient of friction in $[x_0, x_1]$ is smaller/larger than epsilon, the frictional force is/isn’t negligible.
3. The asterisked tasks are the inter-mw-tasks which have the pointers to the microworlds to be transferred to.
4. In MWs, the causal relations between (m2), (m3) and (m4) are explicitly described.

Suppose a learner who has learned ‘uniform motion’ by the intra-mw-task (1) in MW-1 is provided with the inter-mw-task (2*) of MW-1. She/he would be encouraged to transfer to MW-2 because the friction becomes not negligible by the change of physical situation in the task (by accomplishing this task, she/he would learn the ‘decelerated motion’ and ‘frictional force,’ which is the difference between MW-1 and MW-2). Suppose, on the other hand, she/he is provided with the inter-mw-task (3*) of MW-1. She/he would be encouraged to transfer to MW-4 because, in order to accomplish the task, it is necessary to consider the behavioral range (after collision) which is out of consideration in MW-1 (she/he would learn the ‘elastic collision,’ which is the difference between MW-1 and MW-4). In addition, suppose a learner is provided with the inter-mw-task (3*) in MW-2. If she/he use only the knowledge/skills she/he has acquired in MW-2, she/he would get a wrong solution. This error encourages her/him to learn the ‘heat generation’ and ‘melt of the ice,’ that is, to transfer to MW-3. In the similar way, the inter-mw-task (2*) in MW-4 encourages a learner to learn the ‘inelastic collision,’ that is, to transfer to MW-5.

3. Assistance in Microworld-Transition by Parameter Change Rules

There are two types of microworld-transitions: the one which changes the behavioral range of the system to be considered or the viewpoint for the modeling (m4), and the other which (slightly) changes the physical situation of the system (m3). In the former, a learner usually can’t execute the procedure she/he previously learned for getting a solution because the different type of knowledge/skills (model) is required in the new microworld (suppose the transition from MW-1 to MW-4 in Figure 1b, for example). This would sufficiently motivate her/him to transfer to the new microworld. In the latter, on the other hand, a learner often could execute the previous procedure as it is. She/he, therefore, might get a wrong solution because the previous knowledge/skill (model) by itself is irrelevant to the new microworld (suppose the transition from MW-1 to MW-2 in Figure 1b, for example), and she/he might not be aware of the error. This makes it difficult for her/him to transfer to the new microworld.

In such a case, it is necessary to explain why the learner’s solution is wrong compared with the correct solution, in other words, how/why her/his previous model irrelevant to the new situation differs from the ‘right’ model in the situation. Therefore, the model-based explanation is required which relates the difference between the behavior of the wrong and right models with the one between their modeling assumptions (that is, it relates the observable effect of the error with its cause). In this chapter, we show the method for generating such explanation by using a set of ‘parameter change rules.’

The framework of GoM has a set of ‘parameter-change rules’ each of which describes how a model-transition (i.e., the change of modeling assumptions) qualitatively effects on the values of parameters calculated by the models. By using them, it becomes possible to infer the relevant model-transition when the values of parameters calculated by the current model (prediction) are different from the ones measured in the real system (observation). In the framework of GMW, such rules can be used for assisting a learner in microworld-transitions, which are described in the following form:

If the modeling assumptions (m2) change to (m2’), and
the modeling assumptions (m3) change to (m3’)
(and the other modeling assumptions (m4) don’t change)
Then the values of some parameters qualitatively change (increase/steady/decrease)

This rule means that if the model of the physical system (i.e., the physical objects and processes to be considered) changes by the change of physical situation, the values of some parameters of the system increase/steady/decrease. Consider the assistance in transferring from one microworld to the other. First, the parameter change rule which matches them is searched. By using it, the inter-mw-task is identified/generated which asks the (change of) values of those parameters when the physical situation changes. If a learner has difficulty in the task, the explanation is generated which relates the difference between the values calculated by the two models with the difference between their modeling assumptions (i.e., the physical objects and processes to be considered). Thus, the necessity of microworld-transitions can be explained based on the difference between the phenomena she/he wrongly predicted and the ones she/he experienced in the microworld.

(Example-2) Curling-like Problem (2)
We illustrate the two parameter change rules of the GMW in Figure 1b: one is for the transition from MW-1 to MW-2 and the other is for the transition from MW-2 to MW-3. They are described as follows:

PC-Rule-1: If sliding(M_1*, ice), friction(*M_1*, ice) =* μ_1*,* $0 < v_0 < v_0^1$*, not sweep(*$[x_0, x_1]$*), and*
changed(μ_1 *< epsilon =>* μ_1 *> epsilon), and*
changed(consider(uniform motion) => consider(uniformly decelerated motion)), and
considered(frictional force)
Then decrease(velocity(M_1*, x))*

PC-Rule-2: If sliding(M_1*, ice), and*
changed(not sweep($[x_0, x_1]$*) => sweep(*$[x_0, x_1]$*)), and*
considered(heat generation, melt of the ice)
Then change(friction(M_1*, ice) =* μ_1 *=> friction(*M_1*, ice) =* μ_2 *; epsilon <* $\mu_2 < \mu_1$*),*
increase(velocity(M_1*, x), position(*M_1*,* $v_1 = 0$*))*

By using PC-Rule-1, it is inferred that the inter-mw-task (m6)-(2*) of MW-1 is relevant to the transition from MW-1 to MW-2 because it asks the (change of) velocity of M_1 when the coefficient of friction μ_1 increases. By using PC-Rule-2, on the other hand, it is inferred that the inter-mw-task (m6)-(3*) of MW-2 is relevant to the transition from MW-2 to MW-3 because it asks the (change of) position at which M_1 stops when the surface the ice is swept. If a learner has difficulty in these tasks, the model-based explanations are generated by using the information in these rules and microworlds.

4. Assistance in Microworld-Transition by Qualitative Difference Rules

The assistance by parameter change rules is based on the quantitative difference of the behavior of physical systems. That is, what motivates a learner to change the model she/he constructed is the fact that the values of parameters calculated by her/his model differs from the ones observed in the microworld (which is calculated by the ‘right’ model). A learner, however, generally tends to maintain her/his current model (hypothesis). Even when the prediction by her/his model contradicts the observation, she/he often tries to dissolve the contradiction by slightly modifying the model (instead of changing the modeling assumptions) [4]. In addition, quantitative differences sometimes provide insufficient information for the change of modeling assumptions. It would be, therefore, often more effective to use the qualitative/intuitive difference for explaining the necessity of microworld-transitions. In this chapter, we show the method for generating such explanation by using a set of ‘qualitative difference rules’ (which are used complementarily to parameter-change rules).

Each of qualitative difference rules describes how a model-transition effects on the qualitative states/behavior of physical systems calculated by the models (e.g., in Figure 1, the existence of the water (the melted ice made by the frictional heat) in MW-3 qualitatively much differs from the absence of it in MW-2, which is out of the scope of parameter-change rules). They are described in the following form:

If the modeling assumptions (m2) change to (m2’), and
the modeling assumptions (m3) change to (m3’)
(and the other modeling assumptions (m4) don’t change)

Then the qualitative differences of the states/behavior of systems occur

In order to describe these rules, we first classify the differences of the states/behavior between two physical systems from some qualitative viewpoints. We then relate such differences to the ones of modeling assumptions by which they could be caused. In order to derive a set of qualitative difference rules systematically, we execute this procedure based on the qualitative process model [Forbus 84]. The procedure is described in the following two sections.

4.1 Concepts of Difference [12]

The purpose of classifying the behavioral 'differences' of physical systems is to provide a guideline for describing the 'educationally useful' qualitative difference rules, which enable the assistance to motivate a learner as much as possible. When a learner can't explain an observed phenomenon by her/his model, she/he is motivated to modify/change it. The strength of motivation and the relevancy of modification/change would much depend on what kind of difference she/he saw between the observation and her/his prediction. In Figure 1, for example, when a learner sees the water in MW-3, she/he who still uses the model of MW-2 would be surprised because it can't exist by her/his prediction. In addition, the deformation of stones in MW-5 (by the inelastic collision) would surprise a learner who still uses the model of MW-4 because they never deform by her/his prediction. Such differences would motivate a learner much more than the (slight) difference of the velocity of M_1 or the (slight) difference of the energy of stones which might be neglected by her/him. Therefore, the difference in physical objects' existence/absence and the one in physical objects' intrinsic properties (i.e., the classes they belong to) are supposed more effective for motivating a learner because of their concreteness, while the difference in the values of physical quantities are supposed less effective because of their abstractness.

There can appear several/various 'differences' when a physical system behaves contrary to a learner's prediction. Though all of them suggest her/his error more or less, it would be better to choose the 'most effective difference' to be pointed out to her/him[*2]. Therefore, the possible 'differences' and their 'effectiveness' in the behavior of physical systems should be systematically identified and classified. This, in addition, needs to be done in the model-based way because the qualitative difference rules will be described based on this identification/classification.

With this view, we use the qualitative process model [8] because of its reasonable granularity and generality. That is, we regard a physical system and its behavior as a set of physical objects which interact each other through physical processes. The objects are directly/indirectly influenced by the processes and are constrained/changed/generated/consumed. The processes are activated/inactivated when their conditions become true/false. In order to observe the objects in such a system, we introduce the following viewpoints, each of which focuses on:

(v1) how an object exists,
(v2) how a relation between objects is,
(v3) how an object changes through time, and
(v4) how a relation between objects changes through time.

If these are different between in the prediction and in the observation, a learner is supposed to recognize the difference of the behavior.

Based on the viewpoints above, the differences are identified/classified as shown in Figure 2 (they are called 'concepts of difference'). We illustrate some of them (see [12] for more detail):

(d1) the difference about the existence of an object:

If an object exists (or doesn't exist) in the observation which doesn't exist (or exists) in the prediction, it is the difference.

In Figure 1, suppose the behavior of the model in MW-2 is the prediction and the one in MW-3 is the observation, the existence of water (the melted ice by the frictional heat) in the latter is recognized as the difference because it can't exist in the former.

(d2) the difference about the attribute(s) an object has (the object class):

If an object has (or doesn't have) the attribute(s) in the observation which the corresponding object doesn't have (or has) in the prediction, it is the difference. In other words, the corresponding objects in the observation and prediction belong to the different object classes.

In Figure 1, suppose the behavior of the model in MW-2 is the prediction and the one in MW-3 is the observation, the ice in the former belongs to '(purely) mechanical object class' because it doesn't have the attribute 'specific heat,' while the one in the latter belongs to 'mechanical and thermotic object class' because it has the attribute 'specific heat.' Therefore, the ice increasing its temperature or melting in the observation is the difference. In addition, suppose the model in MW-4 is the prediction and the one in MW-5 is the observation, the stones in the former belong to 'rigid object class (the deformation after collision can be

[*2] The 'most effective difference' here means it is the most motive one. Of course, the difference should be also 'suggestive' which means it suggests the way to modify/change a learner's model. This issue is discussed in section 4.2. At present, we are giving priority to motivation in choosing the 'most effective difference,' which could be complemented by other 'more suggestive (but less motive) differences.'

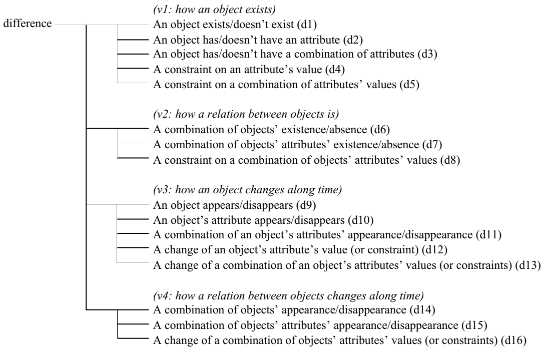

difference
(v1: how an object exists)
An object exists/doesn't exist (d1)
An object has/doesn't have an attribute (d2)
An object has/doesn't have a combination of attributes (d3)
A constraint on an attribute's value (d4)
A constraint on a combination of attributes' values (d5)
(v2: how a relation between objects is)
A combination of objects' existence/absence (d6)
A combination of objects' attributes' existence/absence (d7)
A constraint on a combination of objects' attributes' values (d8)
(v3: how an object changes along time)
An object appears/disappears (d9)
An object's attribute appears/disappears (d10)
A combination of an object's attributes' appearance/disappearance (d11)
A change of an object's attribute's value (or constraint) (d12)
A change of a combination of an object's attributes' values (or constraints) (d13)
(v4: how a relation between objects changes along time)
A combination of objects' appearance/disappearance (d14)
A combination of objects' attributes' appearance/disappearance (d15)
A change of a combination of objects' attributes' values (or constraints) (d16)

Figure 2. Concepts of Differences

ignored),' while the ones in the latter belong to 'elastic object class (the deformation after collision can't be ignored).' Therefore, the deformed stones in the observation are the differences. In both cases, the objects in the observation show 'impossible' natures to a learner.

In general, it would be reasonable to assume the effectiveness of these differences descends from (d1) to (d18) because of their concreteness/abstractness and simpleness/complicatedness. It is of course necessary to identify which differences of them are educationally important and how their effectiveness are ordered depending on each learning domain. The concepts of difference, however, at least provide a useful guideline for describing such knowledge.

4.2 Describing Qualitative Difference Rules

Since the concepts of differences above are identified/classified in model-based way, they can be easily related to the differences of modeling assumptions of the models. That is, each of them can suggest what kind of physical processes, which influence the objects and the constraints on them, are/aren't considered in the models and by what kind of physical situations these processes are/aren't to be considered. This information could be formulated into qualitative difference rules.

The qualitative difference rules are described based on the set of guidelines which are systematically derived from the concepts of differences. We illustrate an example (see [12] for more detail):

(p1) Rules for the differences of the processes which influence (or are influenced by) an object's (dis)appearance:
If an object exists (or doesn't exist) in the observation which doesn't exist (or exists) in the prediction (d1), the followings can be the causes or effects:

1) The process which generates the object is working (or not working) in the former, and is not working (or working) in the latter.
2) The process which consumes the object is not working (or working) in the former, and is working (or not working) in the latter.
3) The influence of the process which generates the object is stronger (or weaker) than the one which consumes the object in the former, and is weaker (or stronger) in the latter.
4) By the existence (or absence) of the object, some process is working (or not working).

Therefore, the following guideline is reasonable:

(Guideline-1)
As for the change of a physical process in (m2) (and the accompanying physical situation in (m3)), the difference about the existence an object can be its effect which is generated/consumed by the process, or can be its cause the existence/absence of which influences the activity of the process.

The qualitative difference rules are used for both identifying/generating inter-mw-tasks and generating model-based explanations, as are the parameter change rules. Especially, when a learner doesn't understand the necessity of microworld-transition, it becomes possible by using them to indicate the qualitative differences of objects which are too surprising to neglect. Since there are usually several qualitative difference rules which match the microworld-transition under consideration, there will be listed several qualitative differences. The effectiveness of them can be estimated based on the concepts of differences and the most effective differences will be chosen.

(Example-3) Curling-like Problem (3)
We illustrate the two qualitative difference rules of the GMW in Figure 1b: one is for the transition from MW-2 to MW-3 and the other is for the transition from MW-4 to MW-5. They are described as follows:

QD-Rule-1: If *sliding(M_1, ice), and*
changed(not sweep([x_0, x_1]) => sweep([x_0, x_1])), and

```
            considered(heat generation, melt of the ice)
       Then appears(water): existence-diff.(d1),
            has-attribute($M_1$, specific-heat): class-diff.(d2)

QD-Rule-2: If collides($M_1$, $M_2$), coefficient-of-restitution($M_1$, $M_2$) = e
            $v_1 > 0$, $v_2 > 0$, and
            changed(e = 1 => 0 < e < 1), and
            changed(consider(elastic collision) => consider(inelastic collision))
       Then deforms($M_1$), deforms($M_2$): class-diff.(d2)
```

By using QD-Rule-1, it is inferred that the inter-mw-tasks are relevant to the transition from MW-2 to MW-3 which focus on the water on the surface of the ice or the increasing temperature of the ice, that is, the differences about the existence of an object or the one about the object class. By using QD-Rule-2, on the other hand, it is inferred that the inter-mw-tasks are relevant to the transition from MW-4 to MW-5 which focus on the deformation of the stones after collision, that is, the differences about the object class. If a learner has difficulty in these tasks, the explanation is generated which relates these differences to the melt process, the heat generation process or inelastic collision process. These rules are, from the viewpoint of motivation, preferred to the parameter change rules matched to these microworld-transitions (the latter identify the tasks which ask the quantitative differences of parameters).

Since there is no qualitative difference rule that match the transition from MW-1 to MW-2, the PC-Rule-1 (which matches it) is used and the inter-mw-task (m6)-(2*) of MW-1 (which asks the quantitative change of the velocity of M_1) is identified as the relevant task.

Concluding Remarks

In this paper, we proposed the GMW framework for assisting a learner's progressive knowledge acquisition in SLEs. Because of its explicit description of microworlds and their differences, the GMW can adaptively navigate a learner in the domain models and generate model-based explanations to assist them. Though the implementation is now ongoing, we believe the GMW greatly helps a learner progressively reconstruct her/his knowledge in a concrete context.

References

[1] Addanki, S., Cremonini, R. and Penberthy, J.S.: Graphs of models, *Artificial Intelligence*, 51, pp.145-177 (1991).
[2] Addanki, S., Cremonini, R. and Penberthy, J.S.: Reasoning about assumptions in graphs of models, *Proc. of IJCAI-89*, pp.1432-1438 (1989).
[3] Burton, R.R., Brown, J.S. & Fischer, G.: Skiing as a model of instruction, In Rogoff, B. & Lave, J. (Eds.), *Everyday Cognition: its development in social context*, Harvard Univ.Press (1984).
[4] Chinn, C.A., Brewer, W.F.: Factors that Influence How People Respond to Anomalous Data, *Proc. of 15th Ann.Conf. of the Cognitive Science Society*, pp.318-323 (1993).
[5] Collins, A. & Gentner, D.: Multiple models of evaporation processes, *Proc. of the Fifth Cognitive Science Society Conference* (1983).
[6] Falkenhainer, B. and Forbus, K.D.: Compositional Modeling: Finding the Right Model for the Job, *Artificial Intelligence*, 51, pp.95-143 (1991).
[7] Fischer, G.: Enhancing incremental learning processes with knowledge-based systems, In Mandl, H. & Lesgold, A. (Eds.), *Learning Issues for Intelligent Tutoring Systems*, Springer-Verlag (1988).
[8] Forbus, K.D.: Qualitative Process Theory, *Artificial Intelligence*, 24, pp.85-168 (1984).
[9] Frederiksen, J. & White, B.: Conceptualizing and constructing linked models: creating coherence in complex knowledge systems, In Brna, P., Baker, M., Stenning, K. & Tiberghien, A. (Eds.), *The Role of Communication in Learning to Model*, pp.69-96, Mahwah, NJ: Erlbaum (2002).
[10] Frederiksen, J. & White, B. & Gutwill, J.: Dynamic mental models in learning science: the importance of constructing derivational linkages among models, *J. of Research in Science Teaching*, 36(7), pp.806-836 (1999).
[11] Goldstein, I.P.: The Genetic Graph: A Representation for the Evolution of Procedural Knowledge, *Int. J. of Man-Machine Studies*, 11, pp.51-77 (1979).
[12] Horiguchi, T. & Hirashima, T.: A simulation-based learning environment assisting scientific activities based on the classification of 'surprisingness', *Proc. of ED-MEDIA2004*, pp.497-504 (2004).
[13] Merrill, M.D.: Instructional Transaction Theory (ITT): Instructional Design Based on Knowledge Objects, In Reigeluth, C.M. (Ed.), *Instructional-Design Theories and Models Vol.II: A New Paradigm of Instructional Theory*, pp.397-424 (Chap. 17), Hillsdale, NJ: Lawrence Erlbaum Associates (1999).
[14] Opwis, K.: The flexible use of multiple mental domain representations, In D. Towne, T. de Jong & H. Spada (Eds), *Simulation-based experiential learning*, pp.77-90, Berlin/New York: Springer (1993).
[15] Stevens, A.L. & Collins, A.: Multiple models of a complex system, In Snow, R., Frederico, P. & Montague, W. (Eds.), *Aptitude, Learning, and Instruction (vol. II)*, Lawrence Erlbaum Associates, Hillsdale, New Jersey (1980).
[16] Towne, D.M.: *Learning and Instruction in Simulation Environments*, Educational Technology Publications, Englewood Cliffs, New Jersey (1995).
[17] Towne, D.M., de Jong, T. and Spada, H. (Eds.): *Simulation-Based Experiential Learning*, Springer-Verlag, Berlin, Heidelberg (1993).
[18] White, B., Shimoda, T.A. & Frederiksen, J.: Enabling students to construct theories of collaborative inquiry and reflective learning: computer support for metacognitive development, *Int. J. of Artifi. Intelli. in Education*, 10(2), pp.151-182 (1999).
[19] White, B. & Frederiksen, J.: Inquiry, modeling, and metacognition: making science accessible to all students, *Cognition and Instruction*, 16(1), pp.3-118 (1998).
[20] White, B. & Frederiksen, J.: ThinkerTools: Causal models, conceptual change, and science education, *Cognition and Instruction*, 10, pp.1-100 (1993).
[21] White, B. & Frederiksen, J.: Causal model progressions as a foundation for intelligent learning environments, *Artificial Intelligence*, 42, pp.99-157 (1990).

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

The Andes Physics Tutoring System: Five Years of Evaluations

Kurt VANLEHN[1], Collin Lynch[1], Kay Schulze[2], Joel A. Shapiro[3], Robert Shelby[4], Linwood Taylor[1], Don Treacy[4], Anders Weinstein[1], and Mary Wintersgill[4]
[1] *LRDC, University of Pittsburgh, Pittsburgh, PA, USA*
[2] *Computer Science Dept., US Naval Academy, Annapolis, MD, USA*
[3] *Dept. of Physics and Astronomy, Rutgers University, Piscataway, NJ, USA*
[4] *Physics Department, US Naval Academy, Annapolis, MD, USA*

Abstract. Andes is a mature intelligent tutoring system that has helped hundreds of students improve their learning of university physics. It replaces pencil and paper problem solving homework. Students continue to attend the same lectures, labs and recitations. Five years of experimentation at the United States Naval Academy indicates that it significantly improves student learning. This report describes the evaluations and what was learned from them.

1 Introduction

Although many students have personal computers now and many effective tutoring systems have been developed, few academic courses include tutoring systems. A major point of resistance seems to be that instructors care deeply about the content of their courses, even down to the finest details. Most instructors are not completely happy with their textbooks; adopting a tutoring system means accommodating even more details that they cannot change.

Three solutions to this problem have been pursued. One is to include instructors in the development process. This lets them get the details exactly how they want them, but this solution does not scale well. A second solution is to include the tutoring system as part of a broader reform with significant appeal to instructors. For instance, the well-know Cognitive Tutors (www.carnegielearning.com) are packaged with an empirically grounded, NCTM-compliant mathematics curriculum, textbook and professional development program. A third solution is to replace grading, a task that many instructors would rather delegate anyway. This is the solution discussed here.

The rapid growth in web-based homework (WBH) grading services, especially for college courses, indicates that instructors are quite willing to delegate grading to technology. In physics, the task domain discussed here, popular WBH services include WebAssign (www.webassign.com), CAPA (www.lon-capa.org/index.html) and Mastering Physics (www.masteringphysics.com). Ideally, instructors still chose their favorite problems from their favorite textbooks, and they may still use innovative interactive instruction during classes and labs. [1] All that changes is that students enter their homework answers on-line, and the system provides immediate feedback on the answer. If the answer is incorrect, the student may receive a hint and may get another chance to derive the answer. Student homework scores are reported electronically to the instructor.

Although WBH saves instructors time, the impact on student learning is unclear. WBH's immediate feedback might increases learning relative to paper-and-pencil homework, or it might increase guessing and thus hurt learning. Although most studies merely report correlations between WBH usage and learning gains, 3 studies of physics instruction have compared learning gains of WBH to those of paper-and-pencil homework (PPH). In the first study, [2] one of 3 classes showed more learning with WBH than PPH. Unfortunately, PPH homework was not collected and graded, but WBH was. It could be that the WBH students did more homework, which in turn caused more learning. In the other studies, [3, 4] PPH problem solutions were submitted and graded, so students in the two conditions solved the roughly the same problems for the same stakes. Despite a large number of students and an impressive battery of assessments, none of the measures showed a difference between PPH students and WBH students. In short, WBH appears to neither benefit nor harm students' learning compared to PPH.

The main goal of the Andes project is to develop a system that is like WBH in that it replaces only the PPH of a course, and yet it increases student learning. Given the null results of the WBH studies, this appears to be a tall challenge. This paper discusses Andes only briefly—see [5] for details. It focuses on the evaluations that test whether Andes increases student learning compared to PPH.

2 The function and behavior of Andes

In order to make Andes' user interface easy to learn, it is as much like pencil and paper as possible. A typical physics problem and its solution on the Andes screen are shown in Figure 1. Students read the problem (top of the upper left window), draw vectors and coordinate axes (bottom of the upper left window), define variables (upper right window) and enter equations (lower right window). These are actions that they do when solving physics problems with pencil and paper.

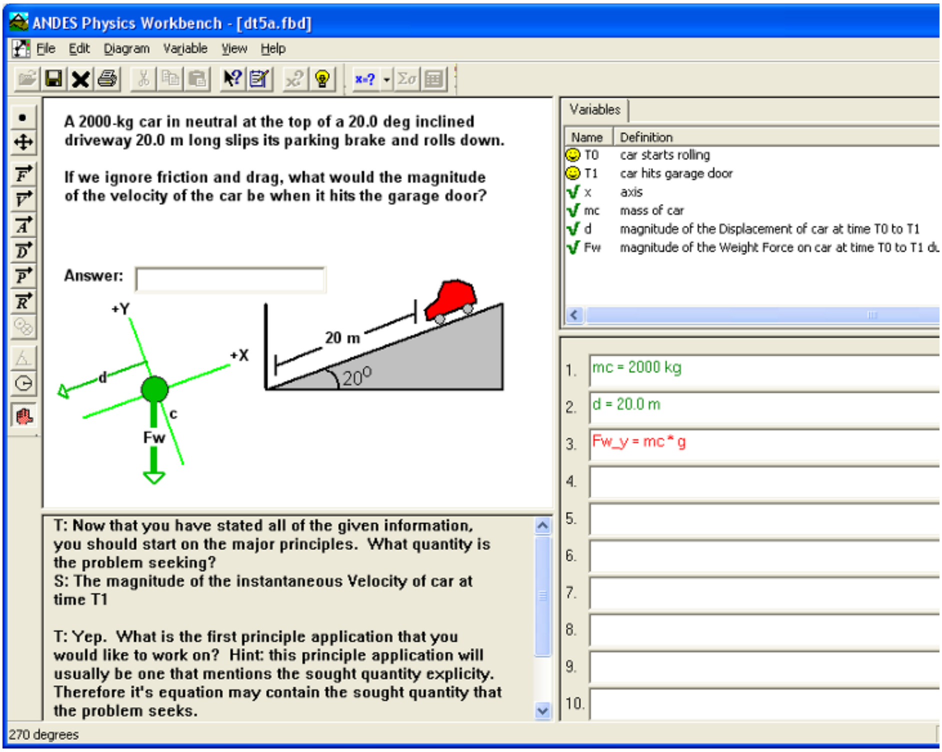
Unlike PPH, as soon as an action is done, Andes gives immediate feedback. Entries are colored green if they are correct and red if they are incorrect. In Figure 1, all the entries are green except for equation 3, which is red.

Also unlike PPH, variables and vectors must be defined before being used. Vectors and other graphical objects are first drawn by clicking on the tool bar on the left edge of Figure 1, then drawing the object using the mouse, then filling out a dialogue box. Filling out these dialogue boxes forces students to precisely define the semantics of variables and vectors. For instance, when defining a force, the student uses menus to select two objects: the object that the force acts on and the object the force is due to.

Andes includes a mathematics package. When students click on the button labeled "x=?" Andes asks them what variable they want to solve for, then it tries to solve the system of equations that the student has entered. If it succeeds, it enters an equation of the form <variable> = <value>. Although physics students routinely use powerful hand calculators, Andes' built-in solver is more convenient and avoids calculator typos.

Andes provides three kinds of help:

- Andes pops up an error messages whenever the error is probably due to lack of attention rather than lack of knowledge. Typical slips are leaving a blank entry in a dialogue box, using an undefined variable in an equation (which is usually caused by a typo), or leaving off the units of a dimensional number. When an error is not recognized as a slip, Andes merely colors the entry red.
- Students can request help on a red entry by selecting it and clicking on a help button. Since the student is essentially asking, "what's wrong with that?" we call this *What's Wrong Help.*


ANDES Physics Workbench - [dt5a.fbd]
File Edit Diagram Variable View Help
+Y
+X
d
c
Fw
20 m
20°
Variables

A 2000-kg car in neutral at the top of a 20.0 deg inclined driveway 20.0 m long slips its parking brake and rolls down.

If we ignore friction and drag, what would the magnitude of the velocity of the car be when it hits the garage door?

Answer:

Name	Definition
T0	car starts rolling
T1	car hits garage door
✓ x	axis
✓ mc	mass of car
✓ d	magnitude of the Displacement of car at time T0 to T1
✓ Fw	magnitude of the Weight Force on car at time T0 to T1 du

1. mc = 2000 kg
2. d = 20.0 m
3. Fw_y = mc * g
4.
5.
6.
7.
8.
9.
10.

T: Now that you have stated all of the given information, you should start on the major principles. What quantity is the problem seeking?
S: The magnitude of the instantaneous Velocity of car at time T1

T: Yep. What is the first principle application that you would like to work on? Hint: this principle application will usually be one that mentions the sought quantity explicity. Therefore it's equation may contain the sought quantity that the problem seeks.

270 degrees

Figure 1: The Andes screen (truncated on the right)

- If students are not sure what to do next, they can click on a button that will give them a hint. This is called *Next Step Help*.

What's Wrong Help and Next Step Help generate a hint sequence that usually has three hints: a pointing hint, a teaching hint and a bottom-out hint. As an illustration, suppose a student who is solving Figure 1 has asked for What's Wrong Help on the incorrect equation *Fw_x = -Fw*cos(20 deg)*. The first hint, which is a pointing hint, is "Check your trigonometry." It directs the students' attention to the location of the error, facilitating self-repair and learning. [6, 7] If the student clicks on "Explain more", Andes gives a teaching hint, namely:

> If you are trying to calculate the component of a vector along an axis, here is a general formula that will always work: Let θV be the angle as you move counterclockwise from the horizontal to the vector. Let θx be the rotation of the x-axis from the horizontal. (θV and θx appear in the Variables window.) Then: V_x = V*cos(θV-θx) and V_y = V*sin(θV-θx).

We try to keep teaching hints as short as possible, because students tend not to read long hints. [8, 9] In other work, we have tried replacing the teaching hints with either multimedia [10, 11]or natural language dialogues. [12] These more elaborate teaching hints significantly increased learning, albeit in laboratory settings.

If the student again clicks on "Explain more," Andes gives the bottom-out hint, "Replace cos(20 deg) with sin(20 deg)." This tells the student exactly what to do.

Andes sometimes cannot infer what the student is trying to do, so it must ask before it can give help. An example is shown in Figure 1. The student has just asked for Next Step Help and Andes has asked, "What quantity is the problem seeking?" Andes pops up a menu or a dialogue box for students to supply answers to such questions. The students' answer is echoed in the lower left window.

In most other respects, Andes is like WBH. Instructors assign problems via email. Students submit their solutions via the web. Instructors access student solutions via a spreadsheet-like gradebook. They can accept Andes' scores for the problems or do their own scoring, and so on.

3 Evaluations

Andes was evaluated in the U.S. Naval Academy's introductory physics class every fall semester from 1999 to 2003. This section describes the 5 evaluations and their results.

Andes was used as part of the normal Academy physics course. The course has multiple sections, each taught by a different instructor. Students in all sections take the same final exam and use the same textbook but different instructors assign different homework problems and give different hour exams, where hour exams are in-class exams given approximately monthly. In sections taught by the authors (Shelby, Treacy and Wintersgill), students were encouraged to do their homework on Andes. Each year, the Andes instructors recruited some of their colleagues' sections as Controls. Students in the Control sections did the same hour exams as students in the Andes section.

Control sections did homework problems that were similar but not identical to the ones solved by Andes students. The Control instructors reported that they required students to hand in their homework, and credit was given based on effort displayed. Early in the semester, instructors marked the homework carefully in order to stress that the students should write proper derivations, including drawing coordinate systems, vectors, etc. Later in the semester, homework was graded lightly, but instructors' marks continued the emphasis on proper derivations. In some classes, instructors gave a weekly quiz consisting of one of the problems from the preceding homework assignment. All these practices encouraged Control students to both do the assignments carefully and to study the solutions that the instructor handed out.

The same final exams were given to all students in all sections. The final exams comprised approximately 50 multiple choice problems to be solved in 3 hours. The hour exams had approximately 4 problems to be solved in 1 hour. Thus, the final exam questions tended to be less complex (3 or 4 minutes each) than the hour exam questions (15 minutes each). On the final exam, students just entered the answer, while on the hour exams, students showed all their work to derive an answer. The hour exam results will be reported first.

3.1 Hour exam results

Table 1 shows the hour exam results for all 5 years. It presents the mean score (out of 100) over all problems on one or more exams per year. In all years, the Andes students scored reliably higher than the Control students with moderately high effect sizes, where effect size defined as (Andes_mean – Control_mean)/Control_standard_deviation. The

Table 1: Hour exam results

Year	1999	2000	2001	2002	2003	Overall
Andes students	173	140	129	93	93	455
Control students	162	135	44	53	44	276
Andes mean (SD)	73.7 (13.0)	70.0 (13.6)	71.8 (14.3)	68.2 (13.4)	71.5 (14.2)	0.22 (0.95)
Control mean (SD)	70.4 (15.6)	57.1 (19.0)	64.4 (13.1)	62.1 (13.7)	61.7 (16.3)	-0.37 (0.96)
P(Andes= Control)	0.036	< .0001	.003	0.005	0.0005	<.0001
Effect size	0.21	0.92	0.52	0.44	0.60	0.61

1999 evaluation had a lower effect size, probably because Andes had few physics problems and some bugs, thus discouraging students from using it. It should probably not be considered representative of Andes' effects, and will be excluded from other analyses in this section.

In order to calculate overall results (rightmost column of Table 1), it was necessary to normalize the exam scores because the exams had different grand means in different years (the grand mean includes all students who took the exam). Each student's exam score was converted to a z-score, where z_score = (score – grand_mean) ÷ grand_standard_deviation. The z-scores from years 2000 through 2003 were aggregated. The overall effect size was 0.61.

The physics instructors recognize that the point of solving physics problems is not to get the right answers but to understand the reasoning involved, so they used a grading rubric for the hour exams that scored the students' work in addition to their answers. In particular, 4 subscores were defined (weights in the total score are shown in parentheses):

- *Drawings:* Did the student draw the appropriate vectors, axes and bodies? (30%)
- *Variable definitions:* Did the student use standard variable names or provide definitions for non-standard names? (20%)
- *Equations:* Did the student display major principle applications by writing their equations without algebraic substitutions and otherwise using symbolic equations correctly? (40%)
- *Answers:* Did the student calculate the correct number with proper units? (10%)

Andes was designed to increase student conceptual understanding, so we would expect it to have more impact on the more conceptual subscores, namely the first 3. Table 2 shows the effect sizes, with p-values from two-tailed t-tests shown in parentheses. Results are not available for 2001. Two hour exams are available for 2002, so their results are shown separately.

There is a clear pattern: The skills that Andes addressed most directly were the ones on which the Andes students scored higher than the Control students. For two subscores, Drawing and Variable definitions, the Andes students scored significantly higher then the Control students in every year. These are the problem solving practices that Andes requires students to follow.

The third subscore, Equations, can also be considered a measure of conceptual understanding. However, prior to 2003, Andes was incapable of discriminating between good and poor usage of equations, so it is not surprising that the Andes and Control students tied on the Equations subscore in years 2000 and 2002. In 2003, Andes gave students warnings and points off on their problem scores if their first use of a major principle was combined algebraically with other equations. Although Andes could have required students to obey this problem solving practice, it only suggested it. This may explain why the Andes students still did no better than the Control students on the Equations subscore in 2003.

The Answers subscore was the same for both groups of students for all years even though the Andes students produced better drawings and variable definitions on those tests. This suggests that the probability of getting a correct answer depends strongly on other skills, such as algebraic manipulation, that are not measured by the more conceptual subscores and not emphasized by Andes. The tied Answer subscores suggest that the

Table 2: Hour exam effect sizes broken down by subscore

Year	**2000**	**2002a**	**2002b**	**2003**	**Average**
Drawings	1.82 (<.001)	0.49 (.003)	0.83 (<.001)	1.72 (<.001)	1.21
Variable definitions	0.88 (<.001)	0.42 (.009)	0.36 (.026)	1.11 (<.001)	0.69
Equations	0.20 (.136)	0.12 (.475)	0.30 (.073)	-0.17 (.350)	0.11
Answers	-0.10 (.461)	-0.09 (.585)	0.06 (.727)	-0.20 (.154)	-0.08

Andes students' use of the equation solving tool did not seem to hurt their algebraic manipulation on the hour exams.

3.2 Final Exam scores

A final exam covers the whole course, but Andes does not. However, its coverage has steadily increased over the years. In 2003, Andes covered 70% of the homework problems in the course. This section reports an analysis of the 2003 final exam data.

In this physics course, engineering and science majors tend to score higher on the final exam than other majors. Unfortunately, there were reliably more engineers among the Andes students than the non-Andes students (p < .0001, 3x2 Chi-squared test). Thus, for each group of majors, we regressed the final exam scores against the students' GPAs. (Of the 931 students, we discarded scores from 19 students with unclassifiable majors or extremely low scores). This yielded three statistically reliable linear models, one for each type of major. For each student, we subtracted the exam score predicted by the linear model from the student's actual score. This residual score represents how much better or worse this student scored compared to the score predicted solely on the basis of their GPA and their major. That is, the residual score factors out the students' general competence. The logic is the same as that used with an ANCOVA, with GPA and major serving as covariates instead of pre-test scores. (This kind of statistical compensation was unnecessary in our analysis of the hour exams, because the distributions of majors and student GPAs did not differ across conditions in any year.)

Using these residual scores, we evaluated Andes' impact on students in each of the 3 groups of majors. As Table 3 indicates, the residual scores of the engineering and science majors were not statistically different with Andes than with paper homework. However, the other majors did learn more with Andes than with paper homework (p=0.013; effect size = 0.52). Over all students, the mean residual scores for Andes students was higher than for non-Andes students (p=0.028; effect size = 0.25).

As though we were gratified to see that Andes students learned more than non-Andes students, we were not surprised that that Andes had little effect on the learning of the engineering and science majors, for two reasons. (1) In many studies, instructional manipulations tend to affect only the less competent students' learning, because highly competent students can usually learn equally well from the experimental and the control instruction [13]. (2) The engineering majors were concurrently taking a course on Statics, which has very similar content to the physics courses. This dilutes the effect of Andes, since it affected only their physics homework and not their Statics homework.

3.3 Comparing Andes to the "benchmark" system

Next we compare our results to results from one of the few large-scaled, controlled field studies of intelligent tutoring systems in the open literature, namely, the evaluation of a combination of an intelligent tutoring system (PAT) and a novel curriculum (PUMP), which is now distributed by Carnegie Learning as the Algebra I Cognitive Tutor. The evaluation was conducted by Koedinger et al. [13]. It is arguably the benchmark against

Table 3: Residual scores on the 2003 final exam

	Engineers	Scientists	Others	All
Andes students	55	9	25	89
Non-Andes students	278	142	403	823
Andes students mean (SD)	0.74 (5.51)	1.03 (3.12)	2.91 (6.41)	1.38 (5.65)
Non-Andes students mean (SD)	0.00 (5.39)	0.00 (5.79)	0.00 (5.64)	0.00 (5.58)
p(Andes=non-Andes)	0.357	0.621	0.013	0.028
Effect size	0.223	0.177	0.52	0.25

which all other tutoring systems should be compared.

Koedinger et al. used both experimenter-defined and standardized tests. Using the experimenter-designed tests, they found effect sizes of 1.2 and 0.7. In our evaluation, the closest matching measures are the Diagram and Variables components of the hour exams, which tap the conceptual skills most directly taught by Andes. Surprisingly, these assessments had exactly the same effect sizes as the Koedinger et al. tests: Diagrams: effect size 1.21; Variables: effect size 0.69.

Koedinger et al. found smaller effect sizes, 0.3, when using multiple-choice standardized tests. The standardized tests most closely match our multiple-choice final exam, where Andes students scored marginally higher than non-Andes students with an effect size of 0.25.

Thus, the Andes evaluations and the Koedinger et al. evaluations have remarkably similar tests and effect sizes. They both have impressive 1.2 and 0.7 effect sizes for conceptual, experimenter-designed tests, and lower effect sizes on standardized, answer-only tests.

The Andes evaluations differed from the Koedinger et al. evaluation in a crucial way. The Andes evaluations manipulated only the way that students did their homework—on Andes vs. on paper. The evaluation of the Pittsburgh Algebra Tutor (PAT) was also an evaluation of the Pittsburgh Urban Mathematics Project curriculum (PUMP), which focused on analysis of real world situations and the use of computational tools such as spreadsheets and graphers. It is not clear how much gain was due to the tutoring system and how much was due to the new curriculum. In our evaluation, the curriculum was not reformed. The gains in our evaluation are a better measure of the power of intelligent tutoring systems *per se*. This is good news for the whole field of intelligent tutoring systems.

4 Conclusions and future work

It appears that we have succeeded in finding a way to use intelligent tutoring systems to help students learn while replacing only their paper-and-pencil homework. Moreover, Andes is probably more effective than existing WBH services, such as WebAssign, CAPA and Mastering Physics. The existing evaluations, which were reviewed in the introduction, suggest that WBH is no more effective than paper-and-pencil homework (PPH), whereas Andes is significantly more effective than PPH. The effect sizes for the open response and multiple choice exams are 0.61 and 0.25, respectively. To be certain that Andes is more effective than WBH, however, one should compare it directly to one of these systems.

We have also shown that Andes' benefits are similar in size to those of the "benchmark" intelligent tutoring system developed by Anderson, Corbett and Koedinger and now distributed by Carnegie Learning. However, Andes' benefits were achieved without attempting to reform the content of the course.

For the immediate future, we have three goals. The first is to help people all over the world use Andes as the U.S. Naval Academy has done, as a homework helper for their courses. Please see www.andes.pitt.edu if you are interested, and please view the training video before trying to use the system.

The second goal is to develop a self-paced, open physics course based on Andes based on mastery learning. We are currently looking for instructors who are interested in developing such a self-paced physics course with us. Please write us if you are interested.

Lastly, the Pittsburgh Science of Learning Center (www.learnlab.org) uses Andes in its physics LearnLab course. A LearnLab course is a regular course that has been heavily instrumented so that investigators can test hypotheses with the same rigor as they would obtain in the laboratory, but with the added ecological validity of a field setting.

5 Acknowledgements

This research was supported by the Cognitive Sciences Program of the Office of Naval Research under grants N00019-03-1-0017 and ONR N00014-96-1-0260, and by NSF under grant SBE-0354420. We gratefully acknowledge the Andes Alumni: Drs. Patricia Albacete, Cristina Conati, Abigail Gertner, Zhendong Niu, Charles Murray, Stephanie Siler, and Ms. Ellen Dugan

6 References

[1] R. R. Hake, "Interactive-engagement vs. traditional methods: A six-thousand student survey of mechanics test data for introductory physics students," *American Journal of Physics*, vol. 66, pp. 64-74, 1998.

[2] R. J. Dufresne, J. P. Mestre, D. M. Hart, and K. A. Rath, "The effect of web-based homework on test performance in large enrollment introductory physics courses," *Journal of Computers in Mathematics and Science Teaching*, vol. 21, pp. 229-251, 2002.

[3] S. W. Bonham, D. L. Deardorff, and R. J. Beichner, "Comparison of student performance using web and paper-based homework in college-level physics," *Journal of Research in Science Teaching*, vol. 40, pp. 1050-1071, 2003.

[4] A. M. Pascarella, "CAPA (Computer-Assisted Personalized Assignments) in a Large University Setting," in *Education*. Boulder, CO: University of Colorado, 2002.

[5] K. Vanlehn, C. Lynch, K. Schultz, J. A. Shapiro, R. H. Shelby, L. Taylor, D. J. Treacy, A. Weinstein, and M. C. Wintersgill, "The Andes physics tutoring system: Lessons learned," *International Journal of Artificial Intelligence and Education*, in press.

[6] G. Hume, J. Michael, A. Rovick, and M. Evens, "Hinting as a tactic in one-on-one tutoring," *Journal of the Learning Sciences*, vol. 5, pp. 23-49, 1996.

[7] D. C. Merrill, B. J. Reiser, M. Ranney, and J. G. Trafton, "Effective tutoring techniques: A comparison of human tutors and intelligent tutoring systems," *The Journal of the Learning Sciences*, vol. 2, pp. 277-306, 1992.

[8] J. R. Anderson, A. T. Corbett, K. R. Koedinger, and R. Pelletier, "Cognitive Tutors: Lessons Learned," *The Journal of the Learning Sciences*, vol. 4, pp. 167-207, 1995.

[9] J.-F. Nicaud, D. Bouhineau, C. Varlet, and A. Nguyen-Xuan, "Towards a product for teaching formal algebra," in *Artificial Intelligence in Education*, S. P. Lajoie and M. Vivet, Eds. Amsterdam: IOS Press, 1999, pp. 207-214.

[10] P. L. Albacete and K. VanLehn, "Evaluation the effectiveness of a cognitive tutor for fundamental physics concepts," in *Proceedings of the Twenty-Second Annual Conference of the Cognitive Science Society*, L. R. Gleitman and A. K. Joshi, Eds. Mahwah, NJ: Erlbaum, 2000, pp. 25-30.

[11] P. L. Albacete and K. VanLehn, "The Conceptual Helper: An intelligent tutoring system for teaching fundamental physics concepts," in *Intelligent Tutoring Systems: 5th International Conference, ITS 2000*, G. Gauthier, C. Frasson, and K. VanLehn, Eds. Berlin: Springer, 2000, pp. 564-573.

[12] C. P. Rose, A. Roque, D. Bhembe, and K. VanLehn, "A hybrid language understanding approach for robust selection of tutoring goals," in *Intelligent Tutoring Systems, 2002: 6th International Conference*, S. A. Cerri, G. Gouarderes, and F. Paraguacu, Eds. Berlin: Springer, 2002, pp. 552-561.

[13] L. J. Cronback and R. E. Snow, *Aptitudes and instructional methods: A handbook for research on interactions.* New York: Irvington, 1977.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

The Politeness Effect: Pedagogical Agents and Learning Gains

Ning WANG[1], W. Lewis JOHNSON[1], Richard E. MAYER[2], Paola RIZZO[3], Erin SHAW[1], Heather COLLINS[2]

[1]Information Sciences Institute, University of Southern California
4676 Admiralty Way, Marina del Rey, CA 90292 USA
[2]Dept. of Psychology, University of California, Santa Barbara
Santa Barbara, CA, 93106-9660 USA
[3]Dept. of Computer Science, University of Rome "La Sapienza"
Via Salaria 113, 00198 Rome, Italy

Abstract. Pedagogical agent research seeks to exploit Reeves and Nass's Media Equation, which holds that users respond to interactive media as if they were social actors. Investigations have tended to focus on the media used to realize the pedagogical agent, e.g., the use of animated talking heads and voices, and the results have been mixed. This paper focuses instead on the manner in which a pedagogical agent communicates with learners, on the extent to which it exhibits social intelligence. A model of socially intelligent tutorial dialog was developed based on politeness theory, and implemented in an agent interface. A series of Wizard-of-Oz studies were conducted in which subjects either received polite tutorial feedback that promotes learner face and mitigates face threat, or received direct feedback that disregarded learner face. The polite version yielded better learning outcomes, and the effect was amplified in learners who expressed a preference for indirect feedback. These results confirm the hypothesis that learners tend to respond to pedagogical agents as social actors, and suggest that research should perhaps focus less on the media in which agents are realized, and place more emphasis on the agents' social intelligence.

Introduction

Researchers have for several years been investigating the potential of pedagogical agents to promote learning. One of the most influential papers in this area was the study by Lester et al. [24] that demonstrated a Persona Effect: that learning was facilitated by an animated pedagogical agent that had a life-like persona and expressed affect. The rationale for this research has been the Media Equation of Reeves and Nass [30], which holds that people tend to respond to interactive media much as they do to human beings. That is, they respond as if the media were social actors.

A number of pedagogical agent investigations have since been conducted, seeking to understand the Persona Effect in more detail, and replicate it in a range of learning domains [17]. The results of these studies have been mixed. For example, the study by Andre et al. [3] showed that animated agents reduce the perceived difficulty of the material being learned, and the study of Bickmore [5] showed that subjects liked an animated agent that responded socially to them, but neither study reported significant learning gains. Moreover, studies by Moreno and Mayer [26] and by Graesser et al. [13] indicated that the agent's voice was the significant contributor to learning outcomes, not the animated persona. Thus the Persona Effect is at best unreliable, and may in fact be a misnomer if the animated persona is not the primary cause of the learning outcomes.

This paper examines a different approach to applying the Media Equation to intelligent tutoring. If as Reeves and Nass suggest learners respond to pedagogical agents as if they were social actors, then the agents' effectiveness should depend upon whether or not they *behave* like social actors. The agents should be socially intelligent, acting in a manner that is consistent with their social role, in accordance with social norms. In fact, human tutors make extensive use of social intelligence when they motivate and support learners [23]. Thus social intelligence in pedagogical agents may be important not just to gain user acceptance, but also to increase the effectiveness of the agent as a pedagogical intervention.

To test this hypothesis, a model of motivational tutorial tactics was developed, based upon politeness theory [18]. A series of Wizard-of-Oz studies were conducted in which subjects either received polite tutorial feedback that promotes learner face and mitigates face threat, or received direct feedback that disregards learner face. The polite version led to improvements in learning outcomes, and the effect was amplified in learners who expressed a preference for indirect feedback. We also observed effects on learner attitudes and motivation [32]. However, we will not describe effects on attitude and motivation in detail here in order to devote as much space as possible to an analysis of the learning outcomes achieved by the polite agent interface.

We term the effect demonstrated here the Politeness Effect. Our results suggest that pedagogical agent research should perhaps place less emphasis on the Persona Effect in animated pedagogical agents, and focus more on the Politeness Effect and related means by which pedagogical agents can exhibit social intelligence in their interactions with learners.

1. The Politeness Theory and Student Motivation

Brown and Levinson [6] have devised a cross-cultural theory of politeness, according to which everybody has a positive and negative "face". Negative face is the want to be unimpeded by others (autonomy), while positive face is the want to be desirable to others (approval). Some communicative acts, such as requests and offers, can threaten the hearer's negative face, positive face, or both, and therefore are referred to as Face Threatening Acts (FTAs). Consider a critique of the learner such as "You did not save your factory. Save it now." There are two types of face threat in this example: the criticism of the learner's action is a threat to positive face, and the instruction of what to do is a threat to negative face.

Speakers use various politeness strategies to mitigate face threats, according to the severity, or "weightiness", of the FTA. In the above case ("You did not save your factory. Save it now."), the tutor could omit the criticism of the learner and focus on the suggested action, i.e., to save the factory. Alternatively the tutor could perform the face-threatening act off record, i.e., so as to avoid assigning responsibility to the hearer. An example of this would be "The factory parameters need saving." The face threat of the instruction can be mitigated using negative politeness tactics, i.e., phrasing that gives the hearer the option of not following the advice, e.g., "Do you want to save the factory now?" Positive politeness strategies can also be employed that emphasizes common ground and cooperation between the tutor and learner, e.g., "How about if we save our factory now?" Other positive politeness strategies include overt expressions of approval, such as "That is very good".

To investigate the role that politeness plays in learner-tutor interaction, in a previous study [16] we videotaped interactions between learners and an expert human tutor while the students were working with the Virtual Factory Teaching System (VFTS) [12], a web-based learning environment for factory modelling and simulation. The expert tutor's comments tended to be phrased in such a way as to have an indirect effect on motivational factors, e.g., phrasing a hint as a question reinforces the learner's sense of control, since the learner can choose whether or not to answer the question affirmatively. Also, the tutor's comments often reinforced the learner's sense of being an active participant in the problem solving process, e.g., by phrasing suggestions as activities to be performed jointly by the tutor and the learner. We are led to think that tutors may use politeness strategies not only for minimizing the weightiness of face threatening acts, but also for indirectly supporting the student's motivation. For instance, the tutor may use positive politeness for promoting the student positive face (e.g. his desire for successful learning), and negative politeness for supporting the student negative face (e.g. his desire for autonomous learning).

2. Related work

In recent years, the recognition of the importance of affect and motivation on learning has led increasingly to the development of socially-aware pedagogical agents as reflected in the work of del Soldato et al. [11] and de Vicente [10]. Heylen et al. [14] highlight the importance of these factors in tutors, and examine the interpersonal factors that should be taken into account when creating socially intelligent computer tutors. Cooper [9] has shown that profound empathy in teaching relationships is important because it stimulates positive emotions and interactions that favour learning. Baylor [4] has conducted experiments in which learners interact with multiple pedagogical agents, one of which seeks to motivate the learner. Other researchers such as Kort et al. [1, 21], and Zhou and Conati [33] have been addressing the problem of detecting learner affect and motivation, and influencing it. User interface and agent researchers are also beginning to apply the Brown & Levinson model to human-computer interaction in other contexts [8, 25]; see also André's work in this area [2].

Porayska-Pomsta [27] has also been using the Brown & Levinson model to analyse teacher communications in classroom settings. Although there are similarities between her approach and the approach described here, her model makes relatively less use of face threat mitigating strategies. This may be due to the differences in the social contexts being modelled.

3. A Wizard-Of-Oz Experiment For Generating And Evaluating Polite Tutor Interventions

In order to apply the theory by Brown and Levinson to the context of interactions in ITSs, we have realized a computational model of politeness in tutorial dialog [18]. In this model, positive and negative politeness values are assigned beforehand to each natural language template that may be used by the tutor. Such values measure the degree to which a template redresses the student's face. We also assign positive and negative politeness values to the tutor, i.e. the degree to which we want the tutor to address the student's positive and negative face. During each communicative act, the template with the politeness values that is closest to the tutor politeness values is selected and used to produce an utterance. For example, a suggestion to save the current factory description, can be stated either bald on record (e.g., "Save the factory now"), as a hint, ("Do you want to save the factory now?"), as a suggestion of what the tutor would do ("I would save the factory now"), or as a suggestion of a joint action ("Why do not we save our factory now?").

To evaluate the intervention tactics, we created a Wizard-of-Oz experiment system aimed at teaching students how to use the VFTS. The student's and experimenter's interfaces are described in detail in [32, 29]; the Plan Recognition and Focus of Attention modules, that help the experimenter analyze student behavior, are described in [28]. The Wizard-of-Oz interface enables a human tutor to use the politeness model to generate the tutorial dialog for those tactics. To communicate with the student, the tutor selects an item in the student activity window (e.g., "copy_factory") then chooses from among a set of communicative acts associated with the current pedagogical goal (e.g., "indicate action & explain reason" or "tell how to perform action") and generates an intervention. The intervention is sent to the agent window on the student interface. An animation engine [31] produces the gestures and a text-to-speech synthesizer synthesizes speech from the text.

3.1. Method

Fifty-one students participated in the study, including 17 students from USC and 34 students from UCSB. The subjects from USC were either engineering graduate or undergraduate students, and all were male. Subjects from UCSB were mostly undergraduate students from introductory psychology classes. Five students from USC participated in a pilot study, which allowed us to test the experiment set-up. Subjects were randomly assigned to either a Polite treatment or a Direct treatment. In the Polite treatment, positive and negative politeness values varied randomly in a moderate to high range, causing the tutor to use politeness in a variety of ways both in giving hints and in providing feedback. In the Direct treatment, positive and negative politeness values were fixed at minimum values, forcing the tutor to communicate directly and not allowing for mitigation of face threat. In all other respects the two treatments were identical.

Two pre-tests were administered: A Background Questionnaire was used to collect information about gender, major and learning style and a Personality Questionnaire was used to measure self-esteem, need for cognition, extroversion and optimism. Personality questions came from the International Personality Item Pool [15]. Two post-test questionnaires were administered as well: A Tutor and Motivation questionnaire was used to evaluate the learner's motivation and perception of the Wizard-of-Oz tutor, and a Learning Outcome questionnaire was used to assess the learner's ability to solve problems on the VFTS.

4. Results

Since the experiment materials and the procedures were identical, we combined the data collected from the experiments carried out in Summer 2004 at USC and in Fall 2004 at UCSB. A two-way analysis of variance (ANOVA) using condition (polite vs. direct) and experiment location (USC vs. UCSB) as between-subject factors showed that there was no significant interaction between condition and experiment location ($F(1, 33)=0.003$, $p=0.957$). Therefore, we focused on comparing the polite and direct conditions using two-tailed t-tests on the combined data (with alpha at $p < .05$).

We excluded a few problematic and extreme cases, due to technical difficulties during the experiment, very extreme personality profiles, and high student ability to complete the task independently. We then grouped the remaining 37 students into two groups: 20 students in Polite and 17 in Direct group, based on the treatment they received. For each group, we calculated the average score of the Learning Outcomes questionnaires and applied Student's t-test to analyze the variance. In this paper, we will only include the analysis on learning gains. The influence on affective factors is not the focus for this paper and will not be included here.

4.1 Overall Polite vs. Direct

Overall, students who received the Polite treatment scored better ($M_{polite}=19.450$, $SD_{polite}=5.6052$) than students who received the Direct treatment ($M_{direct}=15.647$, $SD_{direct}=5.1471$). This is consistent with what we found in our previous study [32]. In the t-test for variance, the difference shows statistical significence ($t(35)= 2.135$, $p=0.040$).

Even though the politeness strategy made an impact on students' learning performance across all students, we're still interested in what group of students is most likely to be influenced by politeness strategies. We grouped students based on their report on the Background and Personality questionnaire, then compared the means between polite and direct groups within students of similar background or personality. The results are presented below.

4.2 Computer skills

From students' self-ratings of their computer skills, we found that almost all students rated their computer skills either average or above average. We then grouped students into 2 groups, 19 with average computer skills and 17 above average (one student with below average computer skill was not included). Overall, students with above average computer skills performed better than students with average computer skills. This may because our test-bed, VFTS, is a relatively complicated computer based teaching system. Better computer skills help students understand the basic concepts of operations in VFTS. But for students with average computer skills, those who received polite treatment (M_{polite}=18.417, SD_{polite}=5.0174) performed better than those who received direct treatment (M_{direct}=14.143, SD_{direct}=3.3877, $t(17)$= 1.993, p=0.063). We did not observe this difference within students with above average computer skills. In this case the tutor, either polite or direct, has less impact on students learning performance. On the other hand, students with poorer computer skills relied more on tutor to help them through the learning task.

4.3 Engineering background

We asked the students whether they work or study in an engineering discipline. Within the students with no engineering background (28 students), we found a major difference between the polite (M_{polite}=18.800, SD_{polite}=5.7966) and direct groups (M_{direct}=14.077, SD_{direct}=4.3677, $t(26)$=2.403, p=0.024). We did not find much difference within engineering students (9 students). VFTS is a system built for Industrial Engineering students. For students who do not work/study in a engineering discipline, such as psychology students, performing tasks in the VFTS could be much more challenging. This is consistent with our hypothesis that students with better ability to perform the task relied less on the tutor.

4.4 Preference for direct help

Direct help are tutor feedbacks that are devoid of any politeness strategy, while Indirect help are the ones that are phrased using politeness strategies. Based on students' preference of direct or indirect help, we grouped them into 3 groups: 15 prefered direct help, 13 prefered indirect and 9 had no preference. For students that prefered direct help or do not have any preference, we did not observe any difference made by the Polite tutor. But for students that specifically reported their preference for indirect help, the Polite tutor made a big difference on their learning performance (M_{polite}=20.429, SD_{polite}=5.7404, M_{direct}=13.000, SD_{direct}=4.5607, $t(11)$= 2.550, p=0.027).

4.5 Frequency of tutor intervention

Tutor attentiveness could be a factor that affected students' learning outcomes. During the experiment, tutor attentiveness was balanced under both experimental conditions. However, how many times of tutor gave feedback to the students depended on the students' need. We grouped students into two groups based on the amount of tutor feedback: 11 students in low and 26 students in average-to-high groups. On average students spent 36 minutes on the VFTS. We considered a low interaction as less than 20 feedbacks during the experiments, while average to high is 20 or more feedbacks. Our hypothesis is that when the number of tutor interventions is low, politeness would have less effect on students' learning. The result confirmed our hypothesis. We found that when the tutor's interventions were low, the Polite tutor did not affect students learning as much. But when the tutor's interventions were average to high, the Polite tutor made a big difference (M_{polite}=18.214, SD_{polite}=5.6046, M_{direct}=13.833, SD_{direct}=3.3530, *t*(24)= 2.365, *p*=0.026).

4.6 Personality

We measured 4 personality traits: self-esteem, optimism, need for cognition and extroversion. On self-esteem and optimism, we found our sample distribution is skewed – most subjects have a high self-esteem and are pretty optimistic. We grouped students based on their level of need for cognition and extroversion. On overall learning results, we did not find interaction between these two personality traits and politeness strategy. However, on students' performance on learning difficult concepts, there are some interesting differences between the polite and direct groups.

For the 20 students scored high on extroversion, we found out that polite tutor helped students to learn difficult concepts more than direct tutor (M_{polite}=10.455, SD_{polite}=2.0671, M_{direct}=8.556, SD_{direct}=1.5899, *t*(18)= 2.259, *p*=0.037). Same difference found for 22 students scored high on need for cognition (M_{polite}=10.000, SD_{polite}=1.4832, M_{direct}=8.182, SD_{direct}=2.5226, *t*(20)= 2.061, *p*=0.053). Students with high need for cognition are probably more motivated to learn difficult concepts. Students with high extroversion are more open to discuss their problems with the tutor when trying to understand difficult concepts. This leads us to believe that, when learning materials are relatively challenging, students with either high extroversion or need for cognition are more likely to be influenced by politeness strategies.

4.7 Liking the tutor

On the post-questionnaire, students were asked whether or not they liked the tutor. We grouped students into 2 groups based on their answers: 20 students liked the tutor and 17 did not or had no preference. We did not find statistical significance between polite and direct group within students did not like the tutor or did not have a preference. But within students that reported that they liked the tutor, we found that students who worked with polite tutor performed better than students worked with direct tutor (M_{polite}=20.333, SD_{polite}=5.2628, M_{direct}=15.500, SD_{direct}=4.9570, *t*(18)=2.058, *p*=0.054), especially on learning difficult concepts (M_{polite}=11.083, SD_{polite}=2.6097, M_{direct}=8.375, SD_{direct}=1.7678, *t*(18)=2.559, *p*=0.020). However, whether students like the tutor or not is not as accurate a predictor of learning performance as preference for direct help.

4.8 Desire to work again with the tutor

We also asked students in the post-questionnaire whether or not they would like to work with the tutor again. We grouped students into 2 groups based on their answers: 22 students would like to work with tutor again and 15 did not or had no preference. We did not find statistical significance between polite and direct group within students who would not like to work with the tutor again or did not have a preference. But within students who reported a desire to work with the tutor again, we found that students who worked with the polite tutor performed better on learning difficult concepts than students worked with the direct tutor (M_{polite}=10.917, SD_{polite}=2.7455, M_{direct}=8.500, SD_{direct}=1.5092, *t*(20)= 2.482, *p*=0.022).

5. Discussion and Conclusion

In this paper, we presented the effect of politeness strategies on students' learning performance, which we call the Politeness Effect. Across all students, a polite agent, compared to a direct agent, had a positive impact on students' learning gains. Richer interaction amplified this effect. And for students with need for indirect help or who had lower ability for the task, the polite agent was much more effective than the direct agent. For students with high extroversion or who were more open to communication with the agent, the polite agent helped them better understand difficult concepts. Making students like the agent appeared to help students learn. But it was not the appearance of the agent, but rather the helpfulness and feedback manner adopted by the agent that created the effect.

Tutorial dialogue is certainly not the only place to apply politeness strategies. In our study, we artificially restricted the use of politeness in tutorial interaction to ensure that the polite condition and the direct condition were as similar as possible. In real human-human interaction, people employ a range of additional strategies to build rapport and react empathetically. These strategies have been modelled in other learning domains [5, 19], and could complement the strategies studied here. We did not include them in this particular study because it would have increased the frequency of tutorial interaction, making it harder to tell whether the Politeness Effect was really a consequence of the frequency of interaction rather than the politeness strategies themselves.

The politeness effect goes beyond the engineering training system we demonstrated here. Other studies we have conducted shown that politeness strategies do occur pervasively in other domains such as second language learning [20]. However, more research will have to be done to study their effects on learning outcomes in other domains.

We recommend that developers of intelligent tutors and pedagogical agents examine the tutorial messages that their tutors are generating from a politeness perspective, as politeness may have an impact on the tutors' effectiveness. Meanwhile, more research needs to be done to study how the Politeness Effect applies in other learning contexts, and investigate other aspects of social actor modelling that go beyond the tactics studied here.

6. Acknowledgement

We would like to thank the various people who have contributed to the Social Intelligence Project, including Wauter Bosma, Maged Dessouky, Mattijs Ghijsen, Sander Kole, Kate LaBore, Hyokeong Lee, Helen Pain, Lei Qu, Sahiba Sandhu, and Herwin van Welbergen. This work was supported in part by the National Science Foundation under Grant No. 0121330, and in part by a grant from Microsoft Research. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or any other funder.

References

[1] Aist, G., Kort, B., Reilly, R., Mostow, J., & Picard, R. (2002). Adding Human-Provided Emotional Scaffolding to an Automated Reading Tutor that Listens Increases Student Persistence. ITS'02, Springer, Berlin.

[2] André, E. Rehm, M., Minker, W., Bühner, D. (2004). Endowing spoken language dialogue systems with emotional intelligence. In Proceedings Affective Dialogue Systems 2004. Springer, Berlin

[3] André, E., Rist, T., Müller, J. (1998). Guiding the User Through Dynamically Generated Hypermedia Presentations with a Life-like Character. Intelligent User Interfaces 1998: 21-28

[4] Baylor, A.L., Ebbers, S (2003). Evidence that Multiple Agents Facilitate Greater Learning. International Artificial Intelligence in Education (AI-ED) Conference. Sydney

[5] Bickmore, T (2003). .Relational Agents: Effecting Change through Human-Computer Relationships" PhD Thesis, Media Arts & Sciences, Massachusetts Institute of Technology.

[6] Brown, P., Levinson, S.C. (1987) Politeness: Some universals in language use. Cambridge University Press, New York

[7] Carbonell, J.R. (1970). AI in CAI: An artificial-intelligence approach to computer-assisted instruction. IEEE transactions on man-machine systems, vol. MMS-11. No.4, December 1970

[8] Cassell, J., Bickmore, T. (2003). Negotiated Collusion: Modeling Social Language and its Relationship Effects in Intelligent Agents. User Modeling and User-Adapted Interaction, 13, 1-2 (2003) 89 – 132

[9] Cooper B. (2003). Care – Making the affective leap: More than a concerned interest in a learner's cognitive abilities. International Journal of Artificial Intelligence in Education, 13, 1

[10] De Vicente, A., Pain, H. (2002) Informing the detection of the students' motivational state: An empirical study. In S.A. Cerri, G. Gouardères, F. Paraguaçu (Eds.): ITS'02. Springer, Berlin 933-943

[11] Del Soldato, T., du Boulay, B. (1995). Implementation of motivational tactics in tutoring systems. Journal of Artificial Intelligence in Education, 6, 4 (1995) 337-378

[12] Dessouky, M.M., Verma, S., Bailey, D., Rickel, J. (2001). A methodology for developing a Web-based factory simulator for manufacturing education. IEEE Transactions 33 167-180

[13] Graesser, A. C., Moreno, K., Marineau, J., Adcock, A., Olney, A., & Person, N. (2003). AutoTutor improves deep learning of computer literacy: Is it the dialog or the talking head? In U. Hoppe, F. Verdejo, and J. Kay (Eds.), Proceedings of Artificial Intelligence in Education (pp. 47-.54). Amsterdam: IOS Press.

[14] Heylen, D., Nijholt, A., op den Akker, R., Vissers, M. (2003). Socially intelligent tutor agents. Social Intelligence Design Workshop

[15] International Personality Item Pool: http://ipip.ori.org/ipip/

[16] Johnson, W.L. (2003). Interaction tactics for socially intelligent pedagogical agents. IUI'03: 251-253

[17] Johnson, W.L., Rickel, J.W., & Lester, J.C. (2000). Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments. In International Journal of Artificial Intelligence in Education.

[18] Johnson, W.L., Rizzo, P., Bosma W., Kole S., Ghijsen M., Welbergen H. (2004). Generating Socially Appropriate Tutorial Dialog. In Proceedings of the Workshop on Affective Dialogue Systems (ADS04). Springer, Berlin

[19] Johnson, W.L., LaBore, C., & Chiu, Y.-C. (2004). A pedagogical agent for psychosocial intervention on a handheld computer. AAAI Fall Symposium on Health Dialog Systems. AAAI Press.

[20] Johnson, W.L., Wu, S. and Nouhi, Y. (2004). Socially Intelligent Pronunciation Feedback for Second Language Learning. In proceedings of IUI'04 workshop n Modelling Human Teaching Tactics and Strategies.

[21] Kort B., Reilly R., Picard R.W. (2001). An Affective Model of Interplay between Emotions and Learning: Reengineering Educational Pedagogy – Building a Learning Companion. In ICALT

[22] Lazarus, R.S. (1991). Emotion and adaptation. Oxford University Press, New York

[23] Lepper, M. R., Woolverton, M., Mumme, D. L., & Gurtner, J. (1993). Motivational techniques of expert human tutors: Lessons for the design of computer-based tutors. In S. P. Lajoie & S. J. Derry (Eds.), Computers as cognitive tools (pp. 75-105). Hillsdale, NJ: Erlbaum.

[24] Lester, J. C., Converse, S. A., Kahler, S. E., Barlow, S. T., Stone, B. A., Bhogal, R. S. (1997). The persona effect: Affective impact of animated pedagogical agents. In CHI '97. 359-366

[25] Miller C. (ed.) (2002) Etiquette for Human-Computer Work. Papers from the AAAI Fall Symposium. AAAI Technical Report FS-02-02

[26] Moreno, R. & Mayer, R. E. (2000). Meaningful design for meaningful learning: Applying cognitive theory to multimedia explanations. ED-MEDIA 2000 Proceedings (pp. 747-752) Charlottesville, VA: AACE Press

[27] Porayska-Pomsta, K. (2004). Influence of Situational Context on Language Production. Ph.D. thesis. University of Edinburgh

[28] Qu, L., Wang, N., Johnson, W.L. (2004). Pedagogical Agents that Interact with Learners. In Proceedings of Workshop on Balanced Perception and Action in ECAs, Intel. Conference on Autonomous Agent & Multiagent Systems. New York

[29] Rizzo P., Shaw E., Lee H., Johnson W.L., Wang N., Mayer R. (2005) A Semi-Automated Wizard of Oz Interface for Modeling Tutorial Strategies. In Proceedings of User Modeling 2005, in press.

[30] Reeves, B., Nass, C. (1996). The media equation. Cambridge University Press, New York

[31] Shaw, E., LaBore, C., C., Y.-C., & Johnson, W.L. (2004). Animating 2D digital puppets with limited autonomy. Proceedings of the Smart Graphics Symposium, Banff, AL.

[32] Wang, N., Johnson, W.L., Rizzo P., Shaw, E. and Mayer, R.E. (2005). Experimental Evaluation of Polite Interaction Tactics for Pedagogical Agents. In Proceedings of IUI'05.

[33] Zhou X., Conati C. (2003). Inferring User Goals from Personality and Behavior in a Causal Model of User Affect. In Proceedings of IUI'03.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Towards Best Practices for Semantic Web Student Modelling

Mike WINTER, Christopher BROOKS[1] and Jim GREER[2]
{mfw127, cab938}@mail.usask.ca[1]
greer@cs.usask.ca[2]
ARIES Laboratory, Department of Computer Science,
Saskatoon, Saskatchewan, Canada S7N 5A9

Abstract. Semantic Web applications offer great potential to student modellers who have traditionally struggled with issues of re-use, portability and tight coupling with learning applications. In this paper, we describe our use of ontology languages and e-learning standards to develop a loosely coupled and portable student modelling architecture used in a large-scale, distributed production learning environment. [1]

Introduction

Student modelling systems face a set of challenges when trying to model student activity on real e-learning systems. The collection of student modelling data is time-consuming and requires the development of data structures to represent student activities within the applications of interest. Once student data is collected, it must be converted into a format compatible with knowledge representation and reasoning systems to function as the input for various adaptive systems. Faced with these requirements, student modelling data is often stored in proprietary, hard-to-access formats that don't encourage reuse or distributed study. Additionally, student modelling systems are often tightly coupled with the learning applications they are developed for, rendering them useless when the application is changed or replaced.

Recently, student modelling researchers have begun to adopt technologies, applications and standards from the Semantic Web and e-learning communities to solve the problems mentioned above. Student modellers are developing their domain models and student models using semantic web ontology language such as the Resource Description Framework Schema (RDFS) or Web Ontology Language (OWL) [2][4][13]. Student models developed with a semantic web ontology language have the advantages of formal semantics, easy reuse, easy portability, availability of effective design tools, and automatic serialization into a format compatible with popular logical inference engines. To support loosely coupled student modelling systems, developers are working with e-learning environments that conform to widely accepted e-learning specifications, such as those developed by the IMS Global Learning Consortium[2]. Student modelling systems that are developed using techniques from the Semantic Web and e-learning specifications have the potential for greater relevance and reuse in real learning systems.

[1] Funding for this research was provided by the LORNET grant from the National Science and Engineering Research Council of Canada.

[2] http://www.imsglobal.org/

The University of Saskatchewan Advanced Research in Intelligent Educational Systems (ARIES) laboratory has spent the past year using Semantic Web tools and e-learning specifications to develop a loosely coupled and reusable student modelling architecture. This architecture aggregates student data from multiple e-learning applications that have large amounts of use from real students. The Semantic Web middleware application developed to transport the student data from the e-learning applications to interested researchers has been discussed in previous publications [5][2], so in this paper we focus on the details of effective student modelling using web ontology languages and e-learning specifications and provide recommendations for future ontology-based student modelling projects. The layout of the paper is as follows: Section 1 discusses the use of ontology languages for developing domain models, Section 2 discusses the process of collecting and representing student model data with the use of standards-based e-learning tools and ontology languages while Section 3 gives an overview of the deployment of our student modelling system in a production environment. Finally, Section 4 provides conclusions and discussion on future work.

1. Towards a Best Practice for Ontology-based Student Modelling

1.1 Introduction to Semantic Web Student Modelling

Ontology languages are used to structure and share knowledge, especially for the use of software applications capable of reasoning that require explicit definitions of concepts and the relationships between those concepts. Evolving from various frame-based representation languages, web ontology languages are being developed as part of the World Wide Web Consortium (W3C) Semantic Web project. The W3C's recommended specification for ontology languages is the Web Ontology Language (OWL), which has three different varieties: OWL Lite, OWL DL and OWL Full. Lite to DL to Full, provide different levels of logical expressiveness, with Lite being the least expressive and Full being the most expressive. The logical semantics of OWL DL (and Lite, which is a subset of DL) are based on a description logic, which is a decidable subset of full first-order logic. This means that all inferences available in an OWL DL ontology can be computed. That is not the case for OWL Full, which is not decidable, and has little to no application reasoning support available. For those reasons, most users of OWL strive to keep their ontologies in OWL DL to ensure maximum utility, ease of development and reuse.

An increasing number of student modelling systems using these ontology languages to specify the structure and properties of their associated student models. Typical approaches are found in [4] where OWL ontologies for a human-computer interaction course are automatically generated from a dictionary and then annotated by hand to fully reflect the course content, and in [11] where IMS Learning Design functions are annotated with OWL ontologies representing an individual's domain knowledge. In this section we discuss our experience of developing a set of student model ontologies that maximize the benefits promised by web ontology languages: extensibility, portability, and inferential power.

1.2 Effective Ontology-based Student Modelling

It is not immediately obvious how to construct an effective production student model using existing web ontology languages. We eventually decided to use OWL DL as our ontology language of choice because of its functionality, tool support (in particular, the Protégé[3] development tool) and status as an official W3C recommendation. In terms of the general structure of our student model ontology, our advice is to separate the ontologies into three broad areas: those that that represent student characteristics, those that encapsulate abstract domain knowledge and relationships, and those that model the concrete subset of the domain taught in particular course along with the learning resources available in those courses. This is similar to the approach taken by other researchers who have used ontology languages to develop student modelling systems [13][8]. By loosely coupling the three different types of ontologies, a student modelling application is better able to react to changes in course subject matter, learning material and student type, which often happens on a semester-to-semester basis in practice. Decoupling the abstract domain ontology of an area of study from the ontologies representing the particular topics and learning resources associated with a course is a particular useful practice. The separation allows a generally static domain ontology to be developed that can be reused across multiple courses teaching different aspects or levels of difficulties of the same area of study even as the particular resources and topics in a given class change rapidly.

Separating the general taxonomy of the domain from the particular instances of the topics being taught in a course also provides a solution to a problem facing ontology developers using the OWL DL and OWL Lite variants: representing classes as property values [6]. When developing an ontology using OWL, one cannot have classes as property values (with the exception of the *rdf:Type* property) without moving the ontology into the OWL Full variant, which is not desirable for the reasons stated above. However, a common statement student modellers want to make is of the general form "*user* knows *topic"*. If *topic* is represented in the ontology as a class, then the ontology will be in OWL Full. Separating out the course-specific instances of topics from the classes in the taxonomy that represent the topics in the abstract allows for the ontology to stay in OWL DL without the awkward, maintenance-heavy artifice of some of the Semantic Web Best Practices and Deployment Working Group's solutions to the classes-as-property-values problem [6]. Using such a separation also makes intuitive educational sense for a reusable domain model: if a topic is being taught in a first-year and a third-year course, statements in the ontology saying that students from the respective courses can know the topic at an equal level are not likely accurate (although you could also develop an expressive set of other properties to capture the depth of knowledge learned, as discussed in Section 2).

1.3 Capturing Useful Pedagogical Relationships in the Domain and Course-Specific Ontologies

The most straightforward way in OWL DL to separate the classes that represent the domain model from the instances that represent the topics being taught in a particular course is to use the *subClassOf* property to model the relationships between classes in the abstract domain model and the *instanceOf* property to connect the concrete course topics to the classes in the abstract domain model. Having a domain ontology constructed using these properties provides only generalization/specialization relationships in the general taxonomy and type information for the topic instances of the course. Figure 1 shows a section of our abstract domain model for the HTML domain, which is constructed only with subclass (*is-a)* relationships. Abstract domain models should fully represent all of the topics in a domain so they can be reused between the different courses that teach the domain they represent.

[3] http://protege.stanford.edu/

However, a richer pedagogical vocabulary is needed to accurately represent the educational relationships between the concrete topics in the individual course ontologies.

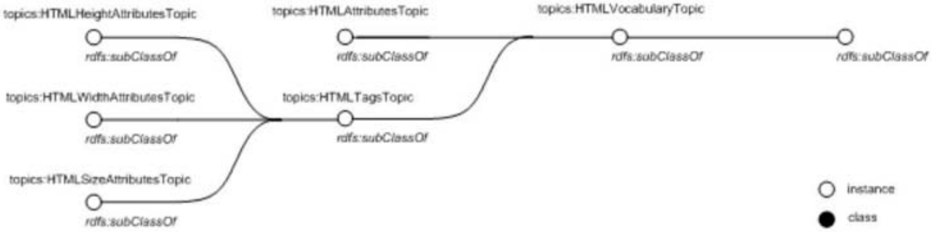

topics:HTMLHeightAttributesTopic
rdfs:subClassOf
topics:HTMLWidthAttributesTopic
rdfs:subClassOf
topics:HTMLSizeAttributesTopic
rdfs:subClassOf
topics:HTMLAttributesTopic
rdfs:subClassOf
topics:HTMLTagsTopic
rdfs:subClassOf
topics:HTMLVocabularyTopic
rdfs:subClassOf
rdfs:subClassOf
○ instance
● class

Figure 1: Fragment of an Abstract Domain Ontology

Our first attempt at using a more expressive educational ontology was to develop an ontology representing the granularity hierarchy formalism, which provides properties representing aggregation and specialization relationships between topics [7]. In the granularity hierarchy ontology, a K-Cluster represents a particular aggregation of topics while an L-Cluster represents a particular specialization of a topic. A topic can have more than one K-Cluster and/or L-Cluster relationship. While the aggregation relationship proved to be a valuable addition to our domain models, we found that granularity hierarchies still did not provide the necessary precision to model all of the different possible relationships between topics in a course, including strong and weak prerequisites.

Another reason to move beyond the granularity hierarchy ontology to describe our course-specific topic ontologies was our larger goal of using standard and widely-accessible tools whenever possible to maximize the portability and extensibility of our student modelling system. There are several widely-used metadata standards that we considered using. An approach taken by Muñoz and de Oliveira in their development of ontologies for the AdaptWeb Knowledge Space project is to model both the domain model and the course topics (which they refer to as a *Content Knowledge Ontology)* with an application profile (instantiated subset) of an RDF binding of the IEEE Learning Object Metadata (LOM) specification [1][3][13]. The LOM specification is a standard developed to describe the metadata associated with a given learning object. It has a rich set of elements to describe learning objects and their use, including *isPartOf* and *hasPrerequisite* properties. However, we decided against using LOM, mainly because it is intended for describing the connections between material learning objects, not the intrinsic pedagogical relationships between the topics presented in a course. Also, the RDF binding of IEEE LOM used by Muñoz and de Oliveira is in OWL Full (Muñoz and de Oliveira used the DAML+OIL ontology language for this particular project, rendering that particular concern irrelevant for them).

The ontologies we decided to use as the basis of our course topic ontologies are from the W3C's Knowledge Organization Systems and the Semantic Web (SKOS) project: SKOS Core [14] and SKOS Extensions [15]. The SKOS family of ontologies was specifically developed to describe taxonomies and classification schemes and thus has an excellent variety of properties to describe the relationship between topics in a course. We developed OWL DL compliant versions of both the Core and Extensions ontologies and used them to develop the topic ontologies of particular course offerings[4]. The Core and Extensions ontologies provide several different variations of aggregation and specialization relationships as well as a class called a *ConceptScheme* that organizes related topics. Our use of the SKOS ontologies in modelling the content of a first-year course teaching HTML is illustrated in Figure 2: we have

[4] http://ai.usask.ca/mums/schemas/2005/01/27/skos-core-dl.owl
http://ai.usask.ca/mums/schemas/2005/01/27/skos-extensions-dl.owl

a *ConceptScheme*, *HTMLConceptScheme*, that represents all of the topics being covered in the course, and all the topics covered in the course are related to the *HTMLConceptScheme* instance by the *inScheme* property (not illustrated in the figure for space reasons). We then model the relationships between topics in the course ontology by using the aggregation and specialization properties provided by SKOS: *cmpt100:HTMLAttributesTopic* is *narrower* than *cmpt100:HTMLVocabularyTopic*, which indicates a specialization relationship, while *cmpt100:HTMLVocabularyTopic* is *relatedHasPart* with *cmpt100:HTMLHyperlinksTopic* which indicates an aggregation relationship between the two topics. All of the topics in the course ontology (represented here by the *cmpt100* namespace) are linked to their respective classes in the abstract domain map by *instanceOf* relationships.

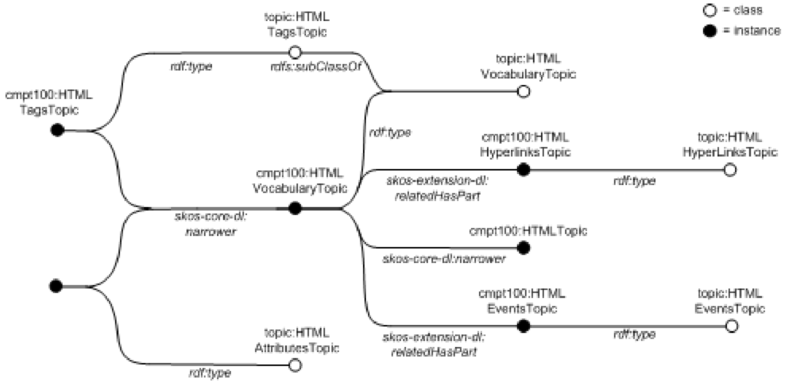


Figure 2: Fragment of an Abstract Domain Ontology with an Attached Concrete Course Ontology that uses SKOS Properties

In this section we demonstrated how we decoupled abstract domain ontologies from course ontologies that only model the topics being taught in particular courses while using the OWL ontology language and staying in its fully logically decidable subset, OWL DL. By using our own DL-compliant binding of the SKOS ontologies we are able to model rich pedagogical relationships between the topics in any given course ontology while still preserving a loosely coupled relationship between the ontologies of different courses that teach the same domain by way of their relationships with OWL classes in the abstract domain map.

2. Modelling Student Competencies and Behaviour

Once the abstract domain and concrete course ontologies are developed, the next step in completing a full student model is to add ontologies about student behaviour and competencies and to develop an effective and portable method to capture student information to populate those ontologies with data. Working towards our goals of maximum reuse and portability, we first examined a number of different standardisation and specification activities taking place in the area of modelling learner competencies. Notable amongst these are the ISO and the IEEE through their work on Public and Private Information (PAPI) for Learners[5], and the IMS Global Learning Consortium and their work on the Learner Information Package (LIP) [11]. These specifications tend to provide containers for learner information as opposed to definitions of what learner information is.

[5] http://jtc1sc36.org/

For instance, both of these schemes allow for the collection of student marks, but nether provides a schema by which to represent student marks. In this way they leave the definition of useful pedagogical content to other specifications, many of which are ill-defined or very general in scope.

Our goal was to develop an ontology that contained an extensive set of educational relationships that could be expressed as ontological properties connecting students with topics in our course-specific topic ontologies discussed in the last section. To this end, we developed an OWL DL ontology[6] that contains the education relationships outlined by Anderson et al. [9]. This variation on Bloom's taxonomy is a two dimensional model that captures both the kind of knowledge gained in a learning experience (e.g. conceptual knowledge, procedural knowledge, etc.) as well as the cognitive processes the student demonstrated in that learning experience (e.g. remembering, understanding, applying, etc.) . We linked in this Anderson-style ontology with our course topic ontologies by making the range of competency statements appear as topics in the topic-course ontologies.

To populate our student competency ontologies with data about real students, we wanted to use standards-compliant e-learning tools so that both our test questions and student competency ontologies could be easily portable. To this end, we developed our test questions to conform to the IMS QTILite specification [12]. This specification describes a data model and XML-based binding for representing questions and tests in a vendor-neutral manner. The model provides ample semantics for representing content, evaluation, and feedback to the learner, but provides no way of associating outcomes of a test with competencies. To connect the test answers to our student competency ontology, we develop a test-specific ‘glue’ ontology that does the work of connecting QTILite answers to statements about student competencies from the Anderson ontology. Figure 3 shows an example segment of a student model that contains a competency statement derived from a QTILite-compliant testing tool.

By adding outcome semantics to individual question/answer pairs, we are able to create fine grained models about a learner's knowledge state. Further, instead of one "correct" answer and many "wrong" answers, we are able to associate any pieces of demonstrated learning with any question/answer pair. While our current tests only associates knowledge statements with one best answer for each question, our loosely-coupled format also allows us to test different levels of knowledge (represented as a collection of answers) within one question. Further, a quick analysis of all of the possible answers for a question, and their associated educational outcomes, allows us to make statements about what knowledge a student has failed to demonstrate in the test, or about the likely misconceptions the student has, given the answer (the classical ‘bug library’).

The final components of our learner model are ontologies that represent the students and the applications they use. Our student ontology is currently very simple, with just the capacity to uniquely identify a student, as we prefer to keep information about students loosely coupled. In the future, however, the ontology may be expanded to include information about a student’s learning style, demographic information or any other factors that are intrinsic to the student. Our application ontologies are more complex, as they model all of the interesting interactions a student can have with our e-learning applications. For example, our message board ontology contains properties to describe a student’s posting of a message with the composition time, the reading of a message with the dwell time, the changing of a category, and many more. These events are not currently translated into any Anderson-style statements about student competency, but they are currently being used for visualization and data mining projects.

[6] http://ai.usask.ca/mums/schemas/2005/01/27/anderson.owl

3. Implementation and Deployment

In this paper, we have emphasized a loosely coupled architecture for ontology-language based student modelling that relies heavily on accepted standards and available tools. This approach was refined over a year-long period of developing ontology-based learner models for students enrolled in a first-year Computer Science course that is offered online at the University of Saskatchewan. Initially, we developed RDFS ontologies that represented every topic in every module of the online course, ranging from the History of Computing to Advanced HTML and Javascript programming. Our initial ontologies contained over 1000 different topics and 1200 granularity hierarchy relationships between the topics [2], as well as around twenty-five QTILite-compliant quizzes embedded into the online courses with over one-hundred questions whose answers were mapped to our topic ontologies.

We immediately ran into problems in the first offering of the course, as the content and organization of the course changed over the semester leaving us unable to update our topic ontologies and questions rapidly enough to permit deployment on the course delivery system. This immediately exposed two problems in our ontology development system. First, our ontology development and maintenance "environment" (Wordpad and gvim) provided no support for rapidly building ontologies or checking their semantic and syntactic correctness. Second, changes in the topic ontology of the course left us with the problem of how to properly maintain the domain knowledge we had invested in modelling, while also storing the knowledge about the differences in the domain material and student behaviour associated with the different offerings of the course. Developing a solution for the second problem led us to the conclusion, helped by the discussion in [13] and [8], that the general domain ontology and the course-specific ontologies should be decoupled, as discussed in Section 1. To solve the first problem, we began to use the Protégé ontology development tool, which is a very mature development platform as well as the core of a large user, plugin and development community. Due in main part to the W3C's recommendation of OWL, a sizable part of the Protégé community is focused on the development of OWL ontologies using the OWL Plugin. A crucial factor in Protégé's popularity is its ability to communicate with logical inference engines, such as Racer, within the development environment. This feature allows developers to check the semantic and inferential correctness of their ontologies as they develop them, and also provides a powerful incentive to stay within the OWL DL language. The ability to use Protégé with the OWL plugin to develop and maintain our ontologies and W3C's recommendation was enough to convince us to convert our ontologies to be in OWL DL.

Currently, we have reduced our initially ambitious goals of trying to focus on maintaining domain, course topic models, and QTILite compliant questions for an entire course, to focusing on two (of twelve) modules within the online course (Introduction to HTML and Programming Languages). This will reduce our overhead as we refine our ontology development process. In addition to the highly structured ontologies and competency data reported in this work, our student modelling repository also contains tens of thousands of ontological statements about student behaviour for hundreds of anonymized undergraduate Computer Science students who use our production e-learning systems, which include the iHelp message board and chat system as well as the online course delivered with the iHelp LCMS [2][5].

4. Conclusions and Future Work

In this paper, we presented recommendations on how to construct an ontology-based learner model, backed by our experience of trying to model students within a real, constantly evolving distributed e-learning environment. We described how we decoupled the abstract domain ontology from the concrete topic ontology representing how the domain is taught in individual offerings of courses. While this approach is somewhat similar to that found in [13] and [8], we instead used the OWL ontology language and the SKOS classification ontology, both endorsed by the W3C, to increase the portability, ease of development, and reuse potential of our learner models. Further, we formalized an educational taxonomy proposed by Anderson et al. to map answers on QTILite-compliant tests in a production online course to statements about student knowledge of topics in our course topic maps, as well as gathering large amounts of information about students' behaviour on various e-learning applications.

In the future, we aim to further refine our ontology-based student models in response to our own experiences and those of the larger student modelling community. With our focus based on ontology-based modelling and the RDF data format, we did not spend large amounts of time analyzing standards strongly associated with XML, such as XML Topic Maps, IMS/IEEE RDCEO and the IMS-LD standard (some discussion on this topic can be found in [10]). Learning to apply these standards in future development would likely be beneficial.

References

[1] Nilsson, M., Palmer, M., Brase, J. *The LOM RDF Binding – Principles and Implementations.* The 3rd Annual Ariadne Conference, 20-21 November 2003.

[2] Winter, Mike, Brooks, Christopher, McCalla, Gord, Greer, Jim and O'Donovan, Peter. *Using Semantic Web Methods for Distributed Learner Modelling*. Proceedings on the Workshop on Using the Semantic Web in E-Learning at the 3rd International Semantic Web Conference, pp. 33-26, 2004.

[3] Nilsson, M. *IEEE Learning Object Metadata RDF Binding* [web page]. May 2001. http://kmr.nada.kth.se/el/ims/metadata.html

[4] Kay, Judy and Lum, Andrew. *Ontologies for Scrutable Learner Modelling in Adaptive E-Learning.* Proceedings of the SWEL Workshop at Adaptive Hypermedia 2004. 2004, pp. 292-301

[5] Brooks, Christopher, Winter, Mike, Greer, Jim and McCalla, Gord. *The Massive User Modelling System (MUMS).* Proceedings of Intelligent Tutoring Systems 2004, pp.635-645.

[6] Noy, Natasha (editor) Stanford University, Stanford, US. – working draft [web page] *Representing Classes as Values on the Semantic Web*. July 21, 2004. http://www.w3.org/TR/swbp-classes-as-values/

[7] McCalla, G. I., Greer, J. E., Barrie, B., and Pospisil, P. *Granularity Hierarchies*, International Journal of Computers and Mathematics with Applications (Special Issue on Semantic Networks), Vol. 23, pp. 363-375, 1992.

[8] Bouzeghoub, Amel, Defude, Bruno, Ammour, Salah, Duitama, John-Freddy and Lecocq Claire. *A RDF Description Model for Manipulating Learning Objects*. Proceedings of the 4th IEEE International Conference on Advanced Learning Technologies. 2004, pp. 81-85,

[9] Anderson, Lorin W. and Krathwol, David R. (eds.). *A Taxonomy for Learning, Teaching and Assessing. A Revision of Bloom's Taxonomy of Education Objectives.* 2001. Addison Wesley Longman Inc.

[10] Dolog, Peter and Nejdl, Wolfgang. *Challenges and Benefits of the Semantic Web for User Modelling.* Proceedings of the Adaptive Hypermedia Workshop at WWW 2003.

[11] Smythe, Colin, Tansey, Frank, Robson, Robby. IMS Learner Information Package Information Model Specification, Version 1.0.

[12] Smythe, Colin, Shepherd, Erik, Brewer, Lane, and Lay, Steve. IMS Question & Test Interoperability QTILite Specification, Version 1.2. February 11, 2002.

[13] Muñoz, Lydia Silva, de Oliveira, Jos᾽é Palazzo Moreira. *Applying Semantic Web Technologies to Achieve Personalization and Reuse of Content in Educational Adaptive Hypermedia Systems*. Proceedings of the SWEL Workshop at Adaptive Hypermedia 2004. 2004, pp. 348-353

[14] Miles, Alistair, Rogers, Nikki and Becket, Dave (editors) W3C [web page] *SKOS Core Ontology.* Feb. 02, 2004. http://www.w3.org/2004/02/skos/core/

[15] Miles, Alistair (editor) W3C [web page] Feb. 02, 2004. http://www.w3.org/2004/02/skos/extensions /

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Critical Thinking Environments for Science Education

Beverly Park WOOLF,[1] Tom MURRAY,[1] David MARSHALL,[1] Toby DRAGON,[1] Kevin KOHLER,[1] Matt MATTINGLY,[1] Merle BRUNO,[2] Dan MURRAY,[3] Jim SAMMONS[3]

Department of Computer Science,[1] University of Massachusetts, Amherst, MA.
School of Natural Science,[2] Hampshire College, Amherst, MA.
Department of Geology,[3] University of Rhode Island, R.I.

Abstract. We have developed a range of critical thinking environments for science education that span several academic content areas, including human biology, geology and forestry. All environments share a methodology, infrastructure and sets of assumptions and tools, which allows them to leverage from the accomplishments and intuitions of the others. These tutors support a student on the Web to be active and engaged, track that student's critical thinking and reason about her knowledge and its own teaching strategies. An *Inquiry Notebook* provides a way to sort, filter and categorize data and justifications and an *Argument Editor* supports argument formation. Students drag and drop evidence to support or refute each argument. A *Coach* provides helpful feedback guided by a database of expert rules, which create the basis for the content-specific analysis of the student's argument.

1. Introduction

We are engaged in several projects to support critical thinking in science education; these projects have both shared and individual goals. The overarching shared goal is to involve students in scientific reasoning, critical thinking and hypothesis generation and thereby generate more responsive and active learning. Individual goals focus on teaching specific academic content knowledge in human biology, geology and forestry. Additionally, each tutor employs consistent elements across disciplines, utilizes common tools and supports intersecting development. This paper describes two inquiry tutors built with this infrastructure and discusses the research approach behind the work.

The inquiry environment, called Rashi,[1] immerses students in problem-based cases and asks them to observe phenomena, reason about them, posit theories and recognize when data does or does not support their hypotheses [1, 2, 3, 4, 5]. Each teaching environment tracks student investigations (e.g., questions, hypotheses, data collection and inferences) and helps the student articulate how evidence and theories are related.

Generic tools, common to all the environments, guide students through ill-structured problem spaces, helping them to formulate questions, frame hypotheses, gather evidence and construct arguments. Tools such as the *Inquiry Notebook* and the *Hypothesis Editor* are used across domains. Domain specific tools, including the *Exam Room* and *Interview Tools* (for human biology), or the *Field Guide* (for forestry) fully engage students in knowledge integration within a specific domain.

[1] Rashi homepage is http://ccbit.cs.umass.edu/Rashihome/

Existing inquiry software presents cases and provides rich simulation-based learning environments and tools for gathering, organizing, visualizing, and analyzing information during inquiry [6, 7, 8, 9, 10]. They support authentic inquiry in the classroom and knowledge sharing, and several tracked and analyzed student data selections and hypotheses. The contribution of this research is to carefully track the reasoning behind student arguments and to critique the student's use of supporting and refuting evidence. The tutor helps students identify weaknesses in their arguments and guides them about how to strengthen arguments during critical thinking. The next two sections describe the Human Biology Inquiry Tutor and then the Geology Tutor.

2. Human Biology Inquiry Tutor

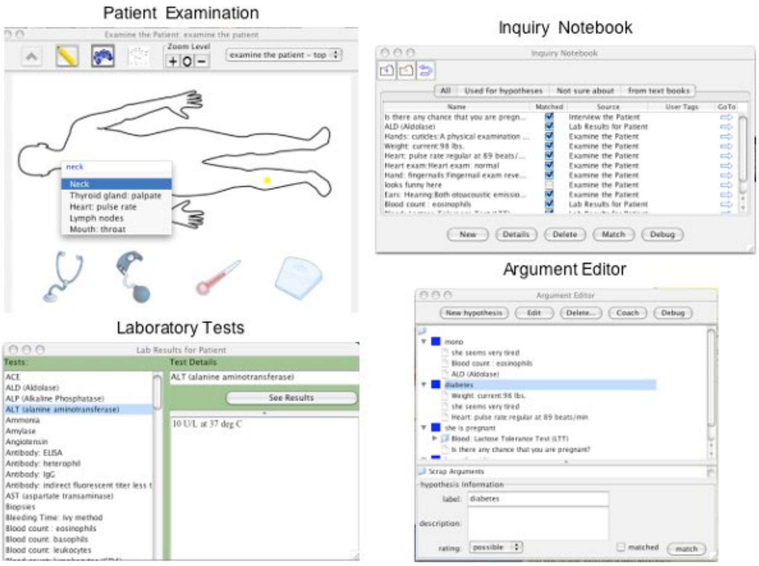
The first domain described is human biology, in which Rashi supports students to evaluate a patient and generate hypotheses about a possible diagnosis.[2] The patient's complaints form an initial set of data from which students begin the diagnostic process, by "interviewing" the patient about symptoms and examining her, Figure 1. Some data is made visible by student action, e.g. asking for chest x-rays, prescribing a certain drug or using a measurement tool. Some data is interpreted for the student (e.g. "x-ray normal"); other data provides raw material and the student interprets it and draws her own conclusions. Six biology cases have been developed, including those for hyperthyroidism, lactose intolerance, food poisoning, diarrhea, mold allergy, and iron deficiency anemia. Hundreds of introductory biology students have used this tutor.

Rashi does not enforce a particular order of student activity, allowing students to move opportunistically from one phase to another. Students read a case description and use tools such as the *Examination Lab* and *Laboratory Examination*, Figure 1, to identify the patient's signs and symptoms. They might interview the patient about her complaints and organize physiological signs, medical history or patient examinations in the *Inquiry Notebook*. They sort, filter and categorize data according to predefined categories and ones that they invent. The site of the observation, e.g., "*Interview Room*" or "*Examination Lab,*" is recorded automatically in the *Inquiry Notebook.* Notebook '*pages'* allow students to create separate spaces for data, as scientists do on separate pages of lab notebooks. A "*scratch pad*" allows a student to record open questions and hypotheses and to identify data that may reveal flaws in a hypothesis. Students search the web for diagnostic material, definitions and interpretations of laboratory results.

Students posit several hypotheses (and other inferences) in the *Argument Editor*, bottom right, Figure 1. They drag and drop data from the *Inquiry Notebook* into the *Argument Editor* to link evidence to support or refute each argument. Arguments can be several levels deep. Structured prompts, reminders and help are student motivated with various stages of inquiry. The student can ask "What do I need to work on?" or "Why is this the wrong hypothesis?" Coaching is based on rules that look for certain conditions in the student's work and provide hints if rules are not met, see Section 4. Currently, the tutor doesn't interrupt the user to provide reminders because that is seen as obtrusive and might potentially slow down the student.

At some point each student makes a final electronic report supporting one hypothesis as the "best." This submission, sent electronically to the teacher, includes data, inferences, hypotheses, justifications, competing hypotheses and arguments from the *Inquiry Notebook* and *Argument Editor*. We are working on a *community-centered* version of the tutor, in which students work in remote groups to brainstorm a list of predictions to resolve a case and each student separately types in possible causes for the observed phenomena.

[2] Human Biology Inquiry Tutor: http://ccbit.cs.umass.edu/Rashihome/projects/biology/index.html



Name	Matched	Source	User Tags	GoTo
Is there any chance that you are pregn...	☑	Interview the Patient		⇨
ALD (Aldolase)	☑	Lab Results for Patient		⇨
Hands: cuticles:A physical examination ...	☑	Examine the Patient		⇨
Weight: current:98 lbs.	☑	Examine the Patient		⇨
Heart: pulse rate:regular at 89 beats/...	☑	Examine the Patient		⇨
Heart exam:Heart exam: normal	☑	Examine the Patient		⇨
Hand: fingernails:Fingernail exam reve...	☑	Examine the Patient		⇨
looks funny here	□	Examine the Patient		⇨
Ears: Hearing:Both otoacoustic emissio...	☑	Examine the Patient		⇨
Blood count : eosinophils	☑	Lab Results for Patient		⇨

- mono
  - she seems very tired
  - Blood count : eosinophils
  - ALD (Aldolase)
- diabetes
  - Weight: current:98 lbs.
  - she seems very tired
  - Heart: pulse rate:regular at 89 beats/min
- she is pregnant
  - Blood: Lactose Tolerance Test (LTT)
  - Is there any chance that you are pregnant?

Scrap Arguments

Hypothesis Information

label: diabetes

description:

rating: possible □ matched

Tests:

ACE
ALD (Aldolase)
ALP (Alkaline Phosphatase)
ALT (alanine aminotransferase)
Ammonia
Amylase
Angiotensin
Antibody: ELISA
Antibody: heterophil
Antibody: IgG
Antibody: indirect fluorescent titer less t
AST (aspartate transaminase)
Biopsies
Bleeding Time: Ivy method
Blood count : eosinophils
Blood count: basophils
Blood count: leukocytes

Test Details

ALT (alanine aminotransferase)

10 U/L at 37 deg C

Figure 1. Human Biology Inquiry Tutor. Diagnosis of the patient begins with an examination and lab tests (left). Examination and interview facts are organized (sometimes automatically) into the *Inquiry Notebook* (top right) and hypotheses are entered into the *Argumentation Editor* (bottom right). In this example, the student has postulated three hypotheses (mono, diabetes and pregnancy) and supported or refuted each with evidence and observations.

Client-server software supports storing individual student data. A simple database houses text entered by the student as well as the *Inquiry Notebook* and *Argument Editor* objects. Intelligence is distributed between the server and client. Java is used to implement visual activities and graphical user interfaces and to reason about the student. This student database is used both to display student work and to reason about it. The reasoning element in Rashi receives data from the student database and compares that with the expert's argument input by the faculty through authoring tools, see Section 5. Rashi searches over both databases to analyze the argument and match student text entries to database objects from the stored expert's argument. The server communicates these results back to the client. The database and all the algorithms for doing the analysis reside in the application and the server is only contacted to store student data. The client side doesn't have a database in any formal sense; though it is primarily the side that analyzes the student's argument, see Section 4. Some portions of the *Coach* reside server side.

3. Geology Inquiry Tutor

This same Rashi inquiry infrastructure supports students using the Geology Tutor to explore a geologic phenomenon and reason about past or future events.[3] In the Fault Recognition Module, Figure 2, students predict where the next earthquake might occur. The module opens with images of a series of large and recurring earthquakes in the San Andreas area of California, U.S.A., bottom Figure 2. The student is asked to relocate a road

[3] The Geology Tutor is at http://ccbit.cs.umass.edu/Rashihome/projects/geology/index.html

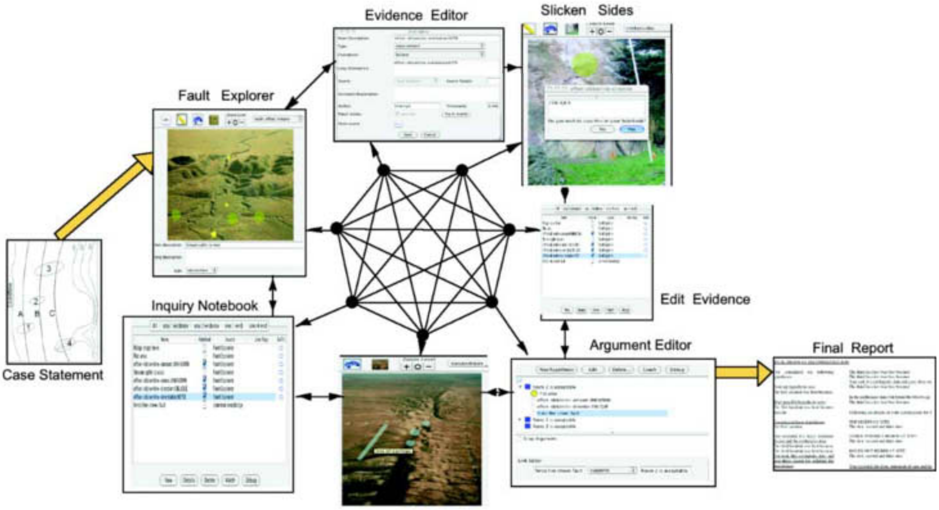


Figure 2. Geology Earthquake Fault Detection Module. The *Case Statement* indicates possible routes for a replacement road, left. Students navigate in any direction through footage of earthquakes, images of fault lines or features such as slicken rock, top right. Observations are noted in the *Inquiry Notebook*, bottom left, arguments made in the *Argument Editor*, bottom right, and a final report is submitted, bottom right.

destroyed by an earthquake, left, Figure 2. Three possible routes are suggested (A, B, or C) each of which pass through combinations of four suspicious areas. As project geologists, students evaluate the area and prepare an engineering report with a best route recommendation. Students from introductory geology courses have used this tutor.

After a student observes an image or activity, she might enter a feature (e.g., lineament or slickenside) into the *Inquiry Notebook.* Elsewhere she might enter inferences (interpretations) of this observation along with supporting reasoning. For example, she might infer that a lineament was an active fault and then support that inference with multiple citations. Hotspots on images provide information, such as: a line of springs parallel to the lineament; a fence offset by 1.3 meters; or a data set that shows that the area was seismographically active. The student is expected to use classroom materials, e.g., to find the relationship between faults and hydrology and to write such observations in her notebook. Finally, the student makes a recommendation (conclusion) of the best place to locate the road supported by observations and inferences. At any point she may ask the *Coach* for help to decide what to do next, or analyze work done so far.

4. Coaching

The *Coach* analyzes a student's argument, compares it to that of the expert and provides useful feedback. Expert knowledge, encoded by a faculty member using the authoring tool, provides a database of expert rules that encapsulates a cohesive argument for each hypothesis. Once an author has made a well-formed expert argument, the *Coach* works effectively to create content-specific analysis of a student's argument.

Remain Domain Independent. Both Rashi and the *Coach* are domain independent and extensible. Expert rules are not specific to a case or domain and do not contain hard-coded domain knowledge. Rules are of two types; either they 1) *support well-formed arguments*, e.g., identify a logical flaw in the student's argument, circular logic, a weakness (lack of supporting data, missing intermediate inferences, etc.) or a missing piece of data, (a location was not visited or lab test not done); or they 2) *shore up weak arguments*, e.g., “You argue that high TSH supports a diagnosis of mononucleosis. I suggest you reconsider this argument.” This second rule type may seem domain dependent, yet it only uses the expert argument from the authoring tool to identify a hypothesis and its evidence. Using the same rule, the *Coach* can then ask the geology student “What supports your hypothesis that route C is preferred?”

Teach good argument structure. By referring to the expert system, the *Coach* teaches students how to make successful inquiry arguments; it guides them to support each argument with sufficient data, making sure an argument is well structured and that no important intermediate steps are skipped. The *Coach* tells a student when to provide proper evidence for an existing hypothesis or when to make a hypothesis/inference for which there is already sufficient data. It shows the student how to support hypotheses with data collected or to collect more data to possibly eliminate hypotheses.

Be Non-Intrusive. Rashi does not enforce a default order for making an argument. If a student is motivated and has a way of doing things, the *Coach* does not intervene. However, if the student is not posing hypotheses or evidential support, the *Coach* will make suggestions and assist her to flesh out details. It will check in the *Inquiry Notebook* to see whether the student has support for an argument and will look in the *Argument Editor* to see that this support is properly connected. The *Coach* will ask a student to connect an argument with its support or to find support if it is missing (e.g., indicate a screen where support can be found).

5. Evaluation and Scaling Up

These inquiry cases have all been evaluated at Hampshire College and the Universities of Massachusetts and Rhode Island with undergraduates as well as middle school science teachers. The biology tutor was evaluated several times in large (500 students) university lecture-based classroom. However, as there was only time to use a few short cases, we consider this evaluation to be a pilot study to test the evaluation instruments. Nevertheless, the results were very encouraging: students quickly learned the software and were able to pose open ended and authentic questions, plan queries and engage in on-line research.

A new evaluation instrument was developed to be sensitive to the small pre-post skill gains that result for short learning interventions and to be more easily scored, see [11]. This instrument measures two types of student learning: 1) *Content questions* ask students in human biology to identify several diagnoses for a set of symptoms and to suggest blood and urine tests; and 2) *Inquiry questions* ask students to critique a set of statements regarding inquiry reasoning and a hypothetical report on a Rashi-like case. The instrument is *item-, recognition- and difference-based.* We only have preliminary data on Rashi use in these cases and it appears that students at the small colleges evidenced gains in content knowledge and no gain in inquiry while students in the large classes showed an increase in inquiry skills and a drop in content performance.

We have also noted significant correlations between a student’s inquiry skill level and some of the Rashi use metrics [11]. In particular, there were significant positive correlations between inquiry skill level and the number of hypotheses posed by a student, the number of arguments, the number of items in the notebook, the number of explanations entered by students, the use of notebook organizing tools and the overall use of Rashi tools. As this is what one would expect, this adds some credence to the ecological validity of the pre-post instrument. As in past formative evaluations of Rashi, the survey did not indicate any significant problems with the software. We interpret these results as supporting the usability of the software and its perceived usefulness.

Interviews, surveys, essay questions, group discussions, and pre-post essay activities have shown that participants were enthusiastic and impressed with the potential of Rashi as an educational tool. Interactivity was seen as a very positive attribute, with the *Patient Examination* feature in biology cited as one of the better components. Students' perception of learning the inquiry process was favorable, Table 1. Half the students felt the experience had taught them how to better approach a comparable inquiry problem.

Since this project is multi-disciplinary and multi-institutional, we need to scale up the usage and coverage of the software. Thus, issues of authoring tool usability and power are critical and perennial. Our experience is that several stakeholders, e.g., faculty and undergraduates, have been able to use our authoring tools to develop new cases in a few weeks after training, see [3]. Experts specify content (images, text, numeric values, etc.), evidential relationships (supports, refutes, etc.) between hypotheses and data and indicate which hypothesis or hypotheses are reasonable conclusions for each case. In one instance, an undergraduate was able to build a biology case in a few weeks as an independent project. She suggested the case, developed medical diagnosis rules and patient signs/symptoms. The case was used with her fellow students the next semester.

Please rate how well you were able to:	%Well/ Very Well
Create hypotheses	53%
Become comfortable with	53%
Learn the content material	47%
Find needed information	47%
Understand the rules for	47%
Use the notebook to	47%
Perform tests	40%
Find the process enjoyable	40%

Table 1. Student reaction to the Human Biology Tutor, Fall '03.

6. The Inquiry Research Strategy

To support active learning for students, who have gown-up with computer and video systems, requires leveraging technology and multimedia to teach domain content and support scientific thinking. We followed a consistent set of *learning* and *pedagogical principles* during development of these tutors as described in this section.

Learning principles. Four learning principles have guided development of this work.

Role-oriented. Students assume roles, e.g., medical personnel responsible for a diagnosis or engineers deciding which location is most secure. Through practice and repetition, students learn the skill of a master in each situation.

Action-oriented. These environments are designed for action and exploration, with experiences structured so that students actively search for a solution. Neither pre-planned knowledge nor explicit course material is delivered.

Goal-oriented. Students are given a goal to pursue while working in a media-rich environment, requiring them to encounter questions and barriers and to generate hypotheses and evidence. Students both learn and rehearse techniques behind critical thinking and also experience the need for domain knowledge while solving a problem.

Interactive and exploratory. Students pursue their own path through the environment. No fixed traversal is enforced. Thus, each learner is self-directed and free to explore and construct knowledge in her own way.

Pedagogical principles. Four pedagogical principles of educational environments have guided this research, based on principles identified in Bransford [12] and expressed in a National Academy of Sciences (NAS) report, *How People Learn* [13]. These principles support delivery of complex learning, domain content and scientific thinking, within authentic and customizable environments. They include:

Knowledge-centered. The tutor knows about domain and student knowledge and can reason about expert rules and a student's arguments.

Learner-centered. The tutor tracks each student's work and responds in the context of that student's reasoning. Students are not treated as blank slates with respect to goals, opinions, knowledge and time.

Assessment-centered. The tutor indicates whether student reasoning is consistent with that of the expert. The *Coach* makes a student's thinking visible and provides chances for the student to review her learning. Assessment is also provided to teachers, in the form of a final report delivered by e-mail, to inform them about student progress.

Community-centered. Currently teams of students work together on a single computer to solve cases. Ultimately people at remote sites will be able to use the tutor to support student collaboration. This latter feature has not been fully implemented.

Producing solid educational material for the Web requires great effort and a large number of resources. Stakeholders, including students, teachers, parents and industries, play a critical role in the process of that material development, with a view towards saving time and resources, as described in this project. All participants need to question the very nature and content of instruction provided on the Web. If the Web is to be worthy of the large investment of time and resources required to impact education, it must provide authentic, flexible and powerful teaching that is responsive to individual students and easy to reproduce and expand.

The set of tutors described in this paper provides a first step in that direction, supporting environments in which students and teachers are involved in authentic problem solving. One of the original dreams behind development of the Web was to provide a realistic mirror (or in fact the primary embodiment) of the ways in which people work and play and socialize [14]. This dream can also be applied to education; thus the Web will become a primary source and environment for education once sufficient intelligent and adaptive teaching materials are available to make education universal and a realistic mirror of anytime and anyplace instruction.

7. Conclusion

This paper described Rashi, a Web-based infrastructure shared by a number of tutors, allowing each to leverage from the accomplishments and intuitions of the others. Rashi supports active and engaging learning on the Web, tracks each student's critical thinking, and reasons about her knowledge and its own teaching strategies, while being open to other resources (Web-sites) and other people (on-line communities). This tutor was not rooted in extensions of what already exists in education, such as lectures or bulletin boards. This paper discussed the shared methodology, infrastructure and tool set.

We observed that students often do not have a great understanding of the inquiry process, but do seem to understand the "scientific method" or a structured method of inquiry learning. Rashi helps students learn the inquiry process, though it doesn't teach it; the tutor provides an environment where inquiry learning is easy to do and intuitive. The student is placed in a situation where she is encouraged to make observations, collect coherent thoughts about these observations and to come up with possible solutions to the questions or problems posed. The *Coach* helps a student learn the inquiry process, not by teaching about the process itself, but by helping the student take part in it. The *Coach* supports students to make hypotheses, find data and use that data to support or refute hypotheses. In sum, Rashi teaches content by providing a problem that requires knowledge of an academic domain to solve. It teaches the inquiry process by involving students in the inquiry process.

8. Acknowledgements

Research on Rashi was funded in part by the U.S. Department of Education, " Expanding a General Model of Inquiry Learning", Fund for the Improvement of Post Secondary Education, Comprehensive Program, #P116B010483, B. Woolf, P.I., and by the National Science Foundation under grant DUE-0127183, "Inquiry Tools for Case-based Courses in Human Biology," M. Bruno, PI and Woolf, Co-PI, and NSF, CCLI #0340864, "On-line Inquiry Learning in Geology," D. Murray, P.I., B. Woolf co-PI.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the funding agencies.

9. References

[1] Woolf, B. P., Marshall, D., Mattingly, M., Lewis, J., Wright, S., Jellison, M & Murray, T. (2003). Tracking student propositions in an inquiry system. In U. Hoppe, F. Berdeho & J. Kay, (Eds.) Artificial Intelligence in Education, Proceedings of AIED 2003, World Conference, IOS Press, pp. 21-28.

[2] Woolf, B. P., Reid, J., Stillings, N., Bruno, M., Murray, D., Reese, P., Peterfreund, A. & Rath, K. (2002) A General Platform for Inquiry Learning, Proceedings of the 6th Int'l Conference on Intelligent Tutoring Systems, Lecture Notes in Computer Science 2363, 681-697, France.

[3] Murray, T., Woolf, B. & Marshall, D. (2004). Lessons Learned from Authoring for Inquiry Learning: A tale of three authoring tools. The International Conference on Intelligent Tutoring Systems, Brazil.

[4] Bruno, M. (2000). Student-active learning in a large classroom. Presented at Project Kaleidoscope 2000 Summer Institute. Keystone, Colorado. http://carbon.hampshire.edu~mbruno/PKAL2000.html

[5] Bruno, M.S. & Jarvis, C. D. (2001). It's Fun, But Is It Science? Goals and Strategies in a Problem-Based Learning Course. *The Journal of Mathematics and Science: Collaborative Explorations, 4*(1): 25-42.

[6] Aleven, V. & Ashley, K. D. (1997). Teaching Case-Based Argumentation Through a Model and Examples: Empirical Evaluation of an Intelligent Learning Environment. In B. du Boulay & R. Mizoguchi (Eds.), Artificial Intelligence in Education, Proceedings of AI-ED 97 World Conference, 87-94. Amsterdam: IOS Press.

[7] Krajcik, J., Blumfeld, P., Marx, R., Bass, K., Fredricks, J. and Soloway, E. (1998). Inquiry in project-based science classrooms: Initial attempts by middle school students. The Journal of the Learning Sciences, 7 (3and4), 313-350.

[8] White, B., Shimoda, T., Frederiksen, J. (1999). Enabling students to construct theories of collaborative inquiry and reflective learning: computer support for metacognitive development. *International J. of Artificial Intelligence in Education, 10*, 151-182.

[9] Suthers, D., Toth, E. & Weiner, A. (1997). An integrated approach to implementing collaborative inquiry in the classroom, *Proceedings of the 2nd Int'l Conference on Computer Supported Collaborative Learning*.

[10] Alloway, G., Bos, N., Hamel, K., Hammerman, T., Klann, E., Krakcik, J., Lyons, D., Madden, T., Margerum-Leys, J., Reed, J., Scala, N., Soloway, E., Vekiri, I., & Wallace, R. (1996). Creating an Inquiry-Learning Environment Using the World Wide Web. *Proceedings of the Int'l Conference of Learning Sciences.*

[11] Murray, T., Rath, K., Woolf, B., Marshall, D., Bruno, M., Dragon, T. & Kohler, K. (2005). Evaluating Inquiry Learning through Recognition Based Tasks, International Conference on AIED, Amsterdam.

[12] Bransford, J.D. (2004). Toward the Development of a Stronger Community of Educators: New Opportunities Made Possible by Integrating the Learning Sciences and Technology, http://www.pt3.org/VQ/html/bransford.html Vision Quest, Preparing Tomorrow's Teachers to Use Technology.

[13] Bransford, J. D., Brown, A. & Cocking, R., (1999). Ed., "How People Learn – Brain, Mind, Experience, and School," National Academy Press, Washington, D.C. 1999.

[14] Berners-Lee, T. (1996). The World Wide Web: Past, Present and Future. IEEE Computer 29(10), 69-77. http://www.w3.org/People/Berners-Lee/1996/ppf.html

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

NavEx: Providing Navigation Support for Adaptive Browsing of Annotated Code Examples

Michael YUDELSON, Peter BRUSILOVSKY
School of Information Sciences, University of Pittsburgh
135 N. Bellefield Ave., Pittsburgh, PA 15260 USA
{peterb, mvy3}@pitt.edu

Abstract. This paper presents NavEx, an adaptive environment for accessing interactive programming examples. NavEx implements a specific kind of adaptive navigation support known as adaptive annotation. The classroom study of NavEx confirmed that adaptive navigation support can visibly increase student motivation to work with non-mandatory educational content. NavEx boosted the overall amount of work and the average length of a session. In addition, various features of NavEx were highly regarded by the students.

Keywords: adaptive environments (web-based and other), motivation and engagement in learning, web-based learning platforms.

1. Introduction

Program examples in the form of small but complete programs play an important role in teaching programming. Program examples help students to understand syntax, semantics and the pragmatics of programming languages, and provide useful problem-solving cases. Experienced teachers of programming-related courses prepare several program examples for every lecture and spend a reasonable fraction of lecture time analyzing these examples. To let the students further explore the examples and use them as models for solving assigned problems, teachers often include the code of the examples in their handouts and even make the code accessible online. Unfortunately, these study tools are not a substitute for an interactive example presentation during the lecture. While the code of the example is still there, the explanations are not. For the students who failed to understand the example in class or who missed the class, the power of the example is lost.

Our system WebEx (Web Examples) developed in 2001 [1] attempted to enhance the value of online program examples by providing *explained examples*. The authoring component of WebEx allowed a teacher to prepare an explained example by adding a written comment for every line of it. The delivery component (see right frame on Figure 1) allowed a student to explore explained examples interactively. Lines with available comments were indicated by green bullets. A click on a bullet opened a comment for the line. This design preserved the structure of an example while allowing the students to selectively open comments for the lines that were not understood. Over the last 4 years we have developed a large set of explained examples for WebEx, used it for several semesters in two different programming-related courses, and run several classroom studies.

In the course of classroom studies of WebEx, the system proved itself as an important course tool. Students rated the system highly, with its ability to support interactive exploration of examples. Many students actively used the system through the course, exploring many examples from different lectures. Yet, a sizeable fraction of students used the system on only a few occasions. Knowing this pattern from our past work on adaptive hypermedia [2], we hypothesized that the students might need some kind of adaptive navigation support that would suggest the most relevant example to explore at any given time. Indeed, with dozens of interactive examples available at the same time, it's not easy to select one to explore. Moreover, WebEx examples were scattered over the course portal with several examples assigned to every lecture. While this organization supported example exploration after a lecture, the abundance of examples made the search for the right example harder.

The experience of ELM-ART [3] demonstrated that the proper adaptive navigation support can significantly increase the amount of student work with a non-mandatory educational content. To gain additional evidence in favor of adaptive navigation support in our context, we solicited student feedback about the need of adaptation in the Spring 2003 study of WebEx. One of the questions in our WebEx questionnaire explained possible adaptive navigation support functionality and asked the students whether this functionality is useful. Almost 70% of respondents (out of 28) rated adaptive navigation support as at least a useful feature (almost 30% rated it as very useful).

This data encouraged us to enhance the original WebEx system with adaptive navigation support. The work on NavEx, an adaptive version of WebEx started in the Fall of 2003 and an early prototype [4] was pilot-tested in Spring 2004. This paper describes the final version of NavEx, which was completed and evaluated in a classroom study in the Fall 2004 semester. The following sections present the interface of NavEx, explain how its adaptive functionality is implemented, and report the results of our classroom study. In brief, the study confirmed positive student attitude toward our adaptive navigation support and demonstrated that one of our specific adaptive navigation support approaches caused impressive growth in system usage.

2. NavEx: The Interface

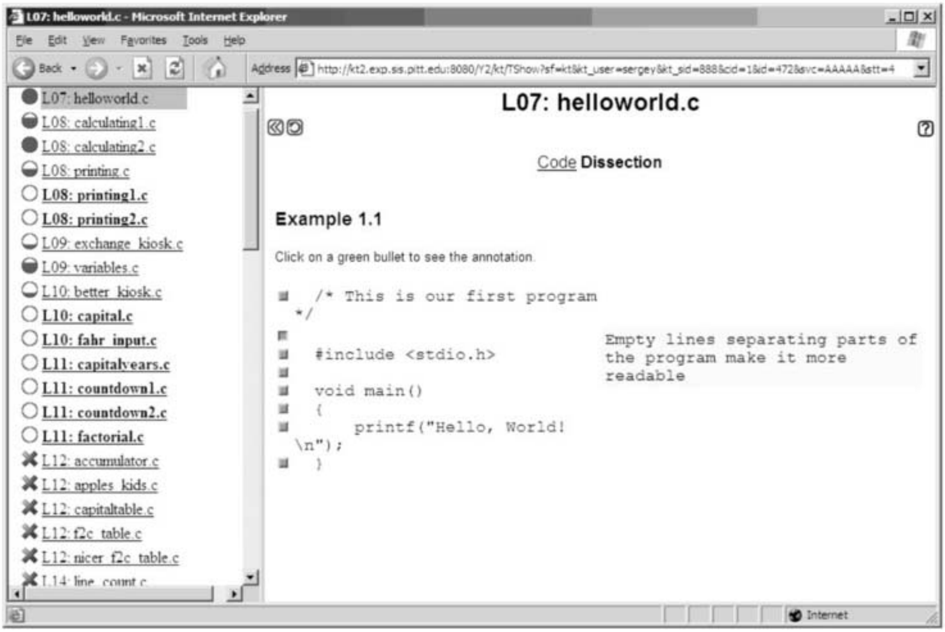
The goal of our NavEx system (Navigation to Examples) is to provide adaptive navigation support in order to access a relatively large set (over 60) of interactive programming examples. Capitalizing on our positive experience with ISIS-Tutor [5], ELM-ART [3] and InterBook [2] we decided to apply a specific kind of adaptive navigation support known as adaptive annotation. With adaptive annotation, a system provides adaptive visual cues for every link to educational content. These visual cues (for example, a special icon or a special anchor font color) provide additional information about the content behind the links helping a student to choose most relevant proper link to follow. One important kind of adaptive annotation pioneered in ISIS-Tutor is zone-based annotation, which divides all educational content into three "zones": 1) sufficiently known, 2) new and ready for exploration, and 3) new, but not-yet-ready. This kind of annotation was later applied in ELM-ART, InterBook, AHA! [6], KBS-HyperBook [7], and many other systems. Another kind of adaptive annotation pioneered in InterBook [3] is progress-based annotation, which shows current progress achieved while working with an educational object. This kind of annotation is currently less popular and is only used in a few systems such as INSPIRE [8].

While the prototype version of NavEx [4] used only zone-based annotation, the current version attempts to combine zone-based and performance-based annotation in a single adaptive icon. The goal of adaptive annotation in NavEx is to provide three types of information to students:

- Categorize examples as being either: ones the student is *ready for* or *not yet ready* to explore;

- Delineate is the student's *progress* within the examples (showing number of explored annotated code lines);
- Emphasize *the most relevant* examples for the student given her past interaction with NavEx or WebEx (all of interaction with WebEx is taken into consideration by NavEx).

The NavEx interface is shown on **Figure 1**. The left side displays a list of annotated links to all the code examples available for a student in the current course. The right side displays the name of the current example, the menu buttons (such as 'reload', 'hide left frame', and 'help'), and the annotated code example.


L07: helloworld.c - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://kt2.exp.sis.pitt.edu:8080/Y2/kt/TShow?sf=kt&kt_user=sergey&kt_sid=888&cid=1&id=472&svc=AAAAA&stt=4
Internet

● L07: helloworld.c
◒ L08: calculating1.c
● L08: calculating2.c
◒ L08: printing.c
○ **L08: printing1.c**
○ **L08: printing2.c**
◒ L09: exchange_kiosk.c
◒ L09: variables.c
◒ L10: better_kiosk.c
○ **L10: capital.c**
○ **L10: fahr_input.c**
○ **L11: capitalyears.c**
○ **L11: countdown1.c**
○ **L11: countdown2.c**
○ **L11: factorial.c**
✖ L12: accumulator.c
✖ L12: apples_kids.c
✖ L12: capitaltable.c
✖ L12: f2c_table.c
✖ L12: nicer_f2c_table.c
✖ L14: line_count.c

L07: helloworld.c

Code Dissection

Example 1.1

Click on a green bullet to see the annotation.

```
■   /* This is our first program
  */
■
■   #include <stdio.h>
■
■   void main()
■   {
■       printf("Hello, World!
  \n");
■   }
```

Empty lines separating parts of the program make it more readable

Figure 1. NavEx interface

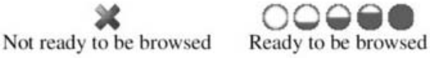


Figure 2. Annotation of the examples

Students click on links in the left frame to select an example and browse annotated code, by clicking again on colored bullets, in order to obtain teacher's comments. Each link to an example in the left frame is supplied with an icon that conveys information about (1) 'readiness' of the student to browse the example, and (2) the student's progress within the example. If the student is 'not ready' to browse the example then a red X bullet is displayed (**Figure 2**). If the student is 'ready' to browse the example then a green round bullet is shown. Depending on the student's progress, the green bullet will be empty, partially or wholly filled. There are 5 discrete progress measures from 0% to 100%, with 25% increments (**Figure 2**). An empty green bullet denotes examples that are available, yet not browsed by the student. The relevance of the example is marked by the font style. If the example is relevant its link is displayed in bold font, otherwise it is in regular font (**Figure**

1). The fact that the example is ‘not ready’ or ‘not recommended’ doesn’t prevent the user from actually browsing it. All of the annotated examples are available for exploration and it is up to a student as to whether to follow the suggestions expressed by annotations or not.

3. NavEx: The Implementation and the Internal Mechanisms

The annotation of examples is compiled, based on the domain model concepts. Each of the examples is indexed with such concepts before it is added to the system. The indexing goes through two stages. First, concepts are extracted from each of the examples by a fully-automatic operation-level parser. Second, for each of the examples, the set of concepts is split into prerequisite concepts and outcome concepts. The splitting algorithm, besides example-concept pairs, requires examples to be grouped by lecture. Indexing algorithms are discussed in more detail in [9]. Supplying each example with two sets of concept - prerequisites and outcomes – plays a two-fold role. First, the concept separation helps to define the learning goals (focus) of the examples in terms of outcomes. Second, concept separation is used for partial ordering of the examples. Thus, an example that has a certain concept as a prerequisite will be placed after an example that has the said concept as outcome.

Once the example is in the system, its annotation for the current user is determined by counting whether or not the current user has mastered the prerequisite concepts. If all of the prerequisite concepts are mastered (or the example simply has no prerequisite concepts) – the example is considered ‘ready to be browsed.’ If the prerequisite concepts are not mastered – the example is marked as ‘not ready to be browsed’. The progress of the student within the example is measured by counting the number of clicks on annotated lines of code example code the user has done with the example.

The relevance of the examples is calculated based on the ‘threshold’ parameter. The ‘threshold’ (calculated for each of the examples individually) is the amount of clicks that has to be done by student for the system to conclude that s/he ‘knows’ the example and declare all of concepts corresponding to example to be mastered. The threshold amount of clicks is calculated as:

$$\textit{threshold} = 0.8 * [\ (\textit{all\_concepts} - \textit{mastered\_concepts}) \,/\, \textit{all\_concepts}\ ] * \textit{all\_clicks}$$

Namely, the total number of clicks possible (for current example) is multiplied by user has to click 80% of the ratio of currently not-mastered concepts (to mastered concepts out of the current example) to all concepts (of the current example). This gives the number of clicks 'left' for user to do and he has to make 80% of those to ‘master’ the example. Only clicks on distinct code lines are counted.total clicks possible. E.g. if there are 10 clicks possible on the lines of the code example and there are 10 concepts assigned to the example: 5 prerequisite (all mastered) and 5 outcomes (none mastered), then the user has to make 0.8 * (5/10) * 10 = 4 clicks to ‘master’ the example. As soon as some concepts are declared mastered the ‘readiness’ of all other examples is recalculated and the mastery of the concepts is propagated further.

The threshold is only used to determine the minimal amount of work the student has to do with the individual example to learn the underlying concepts. The annotation of the examples reflects the absolute amount of student’s work and is not related to the threshold. Since all of the examples share the pool of concepts, it might turn out that at some point there will be one or more examples whose concepts are mastered, yet the student has never browsed those. As mentioned in a previous section, students can browse examples that are annotated as ‘not ready to be browsed’. In extreme cases, the student can browse an example, which contains only concepts that are not yet mastered. To master those concepts while browsing such an example, the student will have to do an extensive amount of clicks, as determined by the threshold.

The NavEx interface is implemented as a server-side solution written in Java. All knowledge and data are stored in a relational database. NavEx is considered to be a value-added service of the KnowledgeTree architecture [10], and uses several protocols, including student modeling and transparent authentication. As a typical value-added service, NavEx resides between E-Learning portals and reusable content objects, providing additional value for teachers and students who use this content through the portal. Unlike other kinds of value-added services, such as annotation services, the value added by NavEx is the ability to adapt to the course goals and student knowledge. With NavEx, teachers can bypass the time-consuming process of selecting examples for each course lecture that meet goal and prerequisite restrictions. Students receive adaptive guidance in selecting examples that are most relevant to their learning goals and knowledge.

4. A Classroom Study of NavEx

A classroom study of NavEx was performed in the context of an undergraduate programming course in the Fall 2004 semester in the School of Information Sciences at the University of Pittsburgh. NavEx was made available to all students taking this course in the second half of the semester, after the midterm exam. There were totally 15 students working with the system. Before the introduction of NavEx the students were able to explore code examples with the original WebEx (i.e., without adaptive guidance) directly through the Knowledge Tree portal. After the introduction, they were able to use both methods of access – with adaptive navigation support through NavEx and without it through the portal and WebEx. User activity collection procedures does not depend on the way students access code examples. Student work with both WebEx and NavEx was equally considered for user modeling.

4.1 Log Analysis

Our main source of data for the study was the user activity log. The log recorded every user click (i.e., every example and code line accessed). Note that the log data gave clear evidence as to whether a student accessed a specific example through NavEx or through WebEx. Since students used WebEx and NavEx in parallel (the use of NavEx was not enforced), a natural way to evaluate the influence of adaptation was to compare the usage profiles of WebEx and NavEx. Analysis of the data showed that NavEx, though introduced late in the course, was considered as a strong alternative to WebEx. After the introduction of NavEx, 56% of example browsing activity was generated by NavEx users. Only 30% of the students didn't use NavEx at all.

Since different students used different "mixtures" of WebEx and NavEx through the course, we decided to assess the added value of the adaptive navigation support by comparing these two systems on a session-by-session basis. A session is counted as a sequence of examples browsed by the student without any sizeable break. The result of this comparison demonstrated clearly the value of adaptive navigation support in increasing the amount of student work with examples.

First, the average session of non-NavEx users was 9.4±0.97 clicks, while NavEx users made an average of 29.6±4.65 clicks per session. That means that navigation support provided by NavEx encouraged students to click on 3.14 times more annotated code lines. Second, the average number of examples browsed per session of non-NavEx users was 1.78±0.15, while NavEx users browsed 2.95±0.46 examples per session. Thus NavEx motivates students to see an average of 1.66 examples more per session. And thirdly, the average length of the non-NavEx user session is 225±33 seconds, while NavEx users have average session length of 885±266 seconds. Hence NavEx keeps students focused on examples 3.9 times longer.

Further evidence can be derived by comparing the example browsing statistics of Fall 2004 semester, when students could use adaptive guidance and Spring 2004 when they could not. Examples set in the Spring 2004 semester had only minor differences from the set of examples available in the Fall 2004 so we can assume that the students had the same external (i.e., tool-independent) motivation to use the tool. The only significant difference was that in the Fall 2004 semester students were able to use NavEx.

The comparison of student activity data of the two semesters demonstrated that the introduction of NavEx boosted the motivation of the students to work more with annotated code examples. The number of code lines accessed per session increased by about 11% from 14.22 in the Spring 2004 semester to 15.8 in the Fall 2004 semester (if we consider only NavEx users the number of clicks per session almost doubled). The average number of line accesses by students over a semester grew by 35% from 323.3 lines in the Spring 2004 semester to 435.9 in the Fall 2004 semester.

Thus, adaptive navigation support succeeded as a tool that encourages the students to work more with examples. It appears that the students were simply more motivated to work with examples when adaptive navigation support was provided. We think that such increase of students' motivations can be attributed to the following reasons. First, navigation support allows students to see 'the big picture' – visualize their current progress with all of their examples and estimate whether the progress they made is enough to move further. Second, since students had all the examples grouped together, they were able to switch from one example to another in fewer clicks and were interested in exploring more examples.

4.2 Subjective Data Analysis

Our secondary source of evaluation data was a non-mandatory questionnaire administered at the end of the term that solicited students' opinions about key features of the system. Out of 15 students in the class, 10 completed the questionnaire.

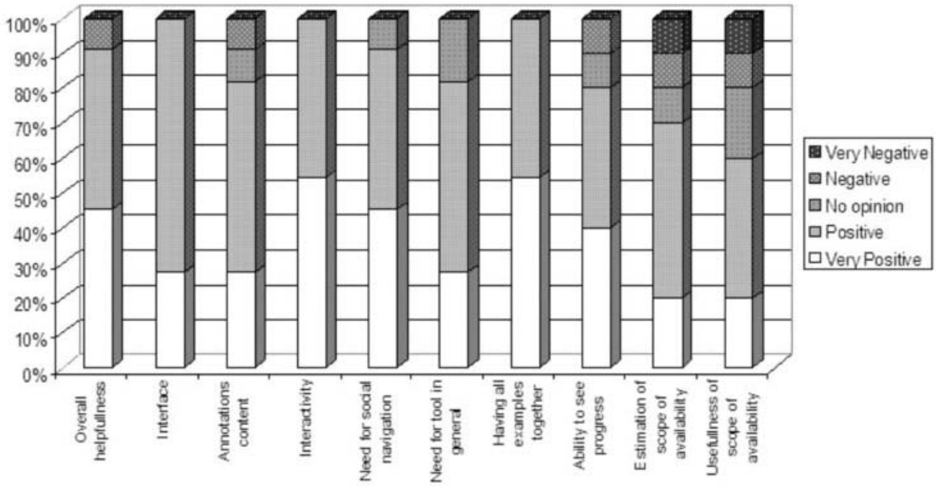


Figure 3. Subjective student evaluation of different features of NavEx

Some of the data obtained from processing the answers is shown in **Figure 3**. As it can be seen, 90% of students considered annotated examples with or without adaptive guidance helpful. 80% percent of students feel positive or strongly positive about the need for such a tool in general. All of the respondents positive or strongly positive evaluated the convenience to have all of the annotated code examples together. 100% of students positively or strongly positively evaluated the interface and the interactive nature of examples.

Two principal features of NavEx: progress indicator and the scope of availability ('readiness') were evaluated positively or strongly positively by a solid fraction of the students (80% for progress indicator and 60-70% for the scope of 'readiness'). The slight downfall of positive response about the scope of 'readiness' of examples' annotation we account to the fact that students started with NavEx in the middle of semester. At the time of their first logon, all of the examples were 'not ready to be browsed', yet at that time students were already familiar with almost half of them and had literally to 'get through' the red X's. Nevertheless, they did appreciate the scope of 'readiness' on the whole.

Students also had a chance to express their suggestions about the future use and development of the system. The idea of students being able to create their own dissections or add their own annotations to the code lines was supported by 70% of respondents (when such activity is an extra credit assignment), and strongly supported by 10% (when such activity is a regular assignment). 90% students expressed strong and very strong support for adding a social navigation feature. A substantial amount of students have also expressed certainty that NavEx should remain as one of the class tools available for students.

5. Summary and Future Work

This paper presented the NavEx system, which provides adaptive navigation support for students accessing interactive program examples. We implemented adaptive navigation support to encourage the students to work more with program examples. Our classroom study confirmed that adaptive navigation support can visibly increase student motivation to work with non-mandatory educational content. NavEx boosted the overall amount of work and the average length of a session. In addition, various features of NavEx were highly regarded by the students. Among two kinds of adaptive navigation support, performance-based annotation was appreciated more than zone-based annotation. However, it may have been influenced by the late introduction of the system.

We plan to perform further studies with NavEx to achieve a better understanding of the value of adaptive navigation support. In addition, we plan to extend the scope of adaptive annotation by providing an annotation of every commented line in an example – not only an example as a whole. To make it possible, we will apply social navigation techniques that we are currently exploring in the course of another project.

References

[1]. Brusilovsky, P. (2001) WebEx: Learning from examples in a programming course. In: W. Fowler and J. Hasebrook (eds.) Proceedings of WebNet'2001, World Conference of the WWW and Internet, Orlando, FL, October 23-27, 2001, AACE, pp. 124-129.

[2]. Brusilovsky, P., Eklund, J. (1998) A study of user-model based link annotation in educational hypermedia. In P. Carlson (ed.) Journal of Universal Computer Science 4 (4), Special Issue on Assessment Issues for Educational Software, 429-448, also available at http://www.iicm.edu/jucs_4_4/a_study_of_user.

[3]. Weber, G., Brusilovsky, P. (2001) ELM-ART: An adaptive versatile system for Web-based instruction. In P. Brusilovsky and C. Peylo (eds.), International Journal of Artificial Intelligence in Education 12 (4), Special Issue on Adaptive and Intelligent Web-based Educational Systems, 351-384, also available at http://cbl.leeds.ac.uk/ijaied/abstracts/Vol_12/weber.html.

[4]. Brusilovsky, P., Yudelson, M., and Sosnovsky, S. (2004) An adaptive E-learning service for accessing Interactive examples. In: J. Nall and R. Robson (eds.) Proceedings of World Conference on E-Learning, E-Learn 2004, Washington, DC, USA, November 1-5, 2004, AACE, pp. 2556-2561.

[5]. Brusilovsky, P. and Pesin, L. (1998) Adaptive navigation support in educational hypermedia: An evaluation of the ISIS-Tutor. Journal of Computing and Information Technology 6 (1), 27-38.

[6]. De Bra, P., Calvi, L. (1998) AHA! An open Adaptive Hypermedia Architecture. In P. Brusilovsky and M. Milosavljevic (eds.), The New Review of Hypermedia and Multimedia 4, Special Issue on Adaptivity and user modeling in hypermedia systems, 115-139.

[7]. Henze, N., Nejdl, W. (2001) Adaptation in open corpus hypermedia. In P. Brusilovsky and C. Peylo (eds.), International Journal of Artificial Intelligence in Education 12 (4), Special Issue on Special Issue on Adaptive and Intelligent Web-based Educational Systems, 325-350, also available at http://cbl.leeds.ac.uk/ijaied/abstracts/Vol_12/henze.html.

[8]. Papanikolaou, K. A., Grigoriadou, M., Kornilakis, H., and Magoulas, G. D. (2003) Personalising the interaction in a Web-based Educational Hypermedia System: the case of INSPIRE. User Modeling and User Adapted Interaction 13 (3), 213-267

[9]. Brusilovsky, P., Sosnovsky, S., Yudelson, M., Chavan, G. (2005) Interactive Authoring Support for Adaptive Educational Systems. In: Proceedings of 12th International Conference on Artificial Intelligence in Education (AIED'2005), Amsterdam, the Netherlands, this volume.

[10].Brusilovsky, P. (2004) KnowledgeTree: A distributed architecture for adaptive e-learning. In: Proceedings of The Thirteenth International World Wide Web Conference, WWW 2004 (Alternate track papers and posters), New York, NY, 17-22 May, 2004, ACM Press, pp. 104-113.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Feedback Micro-engineering in EER-Tutor

Konstantin ZAKHAROV[1]
Antonija MITROVIC[1]
Stellan OHLSSON[2]

[1]Intelligent Computer Tutoring Group, University of Canterbury, Christchurch, New Zealand
[2]Department of Psychology, University of Illinois at Chicago

Abstract: Although existing educational systems are based on various learning theories, these theories are rarely used when developing feedback. Our research is based on the theory of learning from performance errors, which suggests that feedback should provide long and short-term learning advantages through revision of faulty knowledge in the context of learners' errors. We hypothesized that principled, theory-based feedback would have a positive impact on learning. To test the hypothesis we performed an experiment with EER-Tutor, an intelligent tutoring system that teaches database design. The results of the study support our hypothesis: the students who learned from theory-based feedback had a higher learning rate than their peers. We conclude that learning theories should be used to formulate design guidelines for effective feedback.

1. Introduction

Although research in the area of Artificial Intelligence in Education is abundant, there has not been much said about designing effective feedback. Most effort in the area has focused on student modelling, providing problem-solving support and developing pedagogical strategies such as problem selection. Some researchers have investigated the effect of the timing of feedback on learning [6] (i.e. whether immediate feedback is more beneficial than delayed feedback), but advice on how to phrase feedback in order to maximize its impact on learning is hard to find. McKendree [7] compared goal-oriented feedback to pointing out errors and explaining the causes of errors, with the former type of feedback resulting in increased performance and transfer. Most existing educational systems seem to provide what we call *common-sense feedback*. By this, we assume feedback messages generated by system developers based on their intuition and experience. Very often, such feedback tells students what to do, or points out some mistakes in the student's solution.

However, existing educational systems are almost invariably based on some learning theory (such as [1]). Learning theories propose various views on learning, and can be used to develop feedback design guidelines. We believe that in most educational systems feedback is not in line with the underlying learning theory, and propose that principled, theory-based feedback will be more effective that the common-sense one.

In order to test our hypothesis, we performed a study in the context of EER-Tutor, a constraint-based tutor that teaches EER modelling. As is the case with other constraint-based systems developed within the ICTG, EER-Tutor is based on the theory of learning from performance errors, which we briefly overview in Section 2. Section 3 presents the most important features of EER-Tutor, while section 4 describes how feedback messages were re-engineered. The experiment involved two versions of EER-Tutor: the original version which provided common-sense feedback, and a new version providing feedback based on the underlying theory. Section 5 presents the experiment and the results derived from it. Finally, we present the conclusions and the area of future work in the final section.

2. Learning from Performance Errors and Constraint-Based Modeling

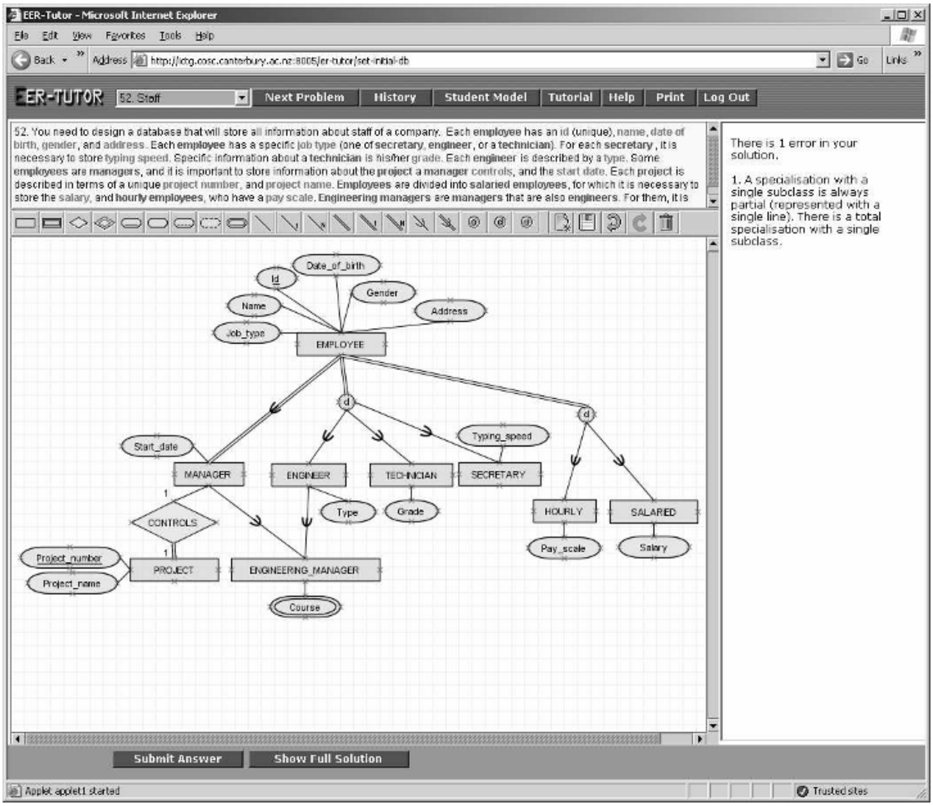
The theory of learning from performance errors [10] proposes that we often make mistakes when performing a task, even when we have been taught the correct way to do it. According to this theory, we make mistakes because the declarative knowledge we have learned has not been internalized in our procedural knowledge, and so the number of decisions we must make while performing the procedure is sufficiently large that we make mistakes. By practicing the task, and catching ourselves (or being caught by a mentor) making mistakes, we modify our procedure to incorporate the appropriate rule that we have violated. Over time, we internalize all declarative knowledge about the task, and so the number of mistakes we make is reduced. The theory views learning as consisting of two phases: *error recognition* and *error correction*. A student needs declarative knowledge in order to detect an error. Only then can the error be corrected so that the solution used is applicable only in situations in which it is appropriate.

Constraint-Based Modeling (CBM) is a student modeling approach [9,8] arising from the above theory. CBM starts from the observation that all correct solutions are similar in that they do not violate any domain principles. CBM is not interested in the exact sequence of states in the problem space the student has traversed, but only in the current state. As long as the student never reaches a state that is known to be wrong, they are free to perform whatever actions they please. Constraints define equivalence classes of problem states. An equivalence class triggers the same instructional action; hence all states in an equivalence class are pedagogically equivalent. It is therefore possible to attach feedback messages directly to constraints. The domain model is a collection of state descriptions of the form: *If <relevance condition> is true, then <satisfaction condition> had better also be true, otherwise something has gone wrong*. In other words, if the student solution falls into the state defined by the relevance condition, it must also be in the state defined by the satisfaction condition in order to be correct.

Constraint-based tutors evaluate student solutions by matching them against the constraint set. Firstly, all relevance patterns are matched against the problem state. Secondly, the satisfaction components of relevant constraints are tested. If a satisfaction pattern matches the state, the constraint is satisfied, otherwise, it is violated. The short-term student model consists of all satisfied and violated constraints. Long-term student model mainly consists of the list of all constrains used by the student and the history of constraint usage.

3. EER-Tutor

Conceptual database modelling, in particular Enhanced Entity-Relationship (EER) modelling [3] is a design task. Goel and Pirolli [4] define generic design (i.e. domain-independent characterization of design tasks) as a radical category, which is described in terms of prototypical examples and some unpredictable variations of them. Design tasks are ill-structured, because their start/goal states and problem-solving algorithms are underspecified. The start state is usually described in terms of ambiguous and incomplete specifications. The problem spaces are typically huge, and operators for changing states do not exist. The goal state is also not clearly stated, but is rather described in abstract terms. There is no definite test to use to decide whether the goal has been attained, and consequently, there is no best solution, but rather a family of solutions. Design tasks typically involve huge domain expertise, and large, highly structured solutions. For these reasons, EER modelling presents a considerable learning challenge. The learner is given an abstract definition of a good solution. In database modelling, a good solution is defined as an EER schema that matches the requirements, and



52. You need to design a database that will store all information about staff of a company. Each employee has an id (unique), name, date of birth, gender, and address. Each employee has a specific job type (one of secretary, engineer, or a technician). For each secretary, it is necessary to store typing speed. Specific information about a technician is his/her grade. Each engineer is described by a type. Some employees are managers, and it is important to store information about the project a manager controls, and the start date. Each project is described in terms of a unique project number, and project name. Employees are divided into salaried employees, for which it is necessary to store the salary, and hourly employees, who have a pay scale. Engineering managers are managers that are also engineers. For them, it is

There is 1 error in your solution.

1. A specialisation with a single subclass is always partial (represented with a single line). There is a total specialisation with a single subclass.

Fig. 1. A screenshot of EER-Tutor

satisfies all the integrity rules of the chosen data model. We have previously showed that CBM is an effective domain and student modelling approach for design tasks [11].

We developed EER-Tutor, a constraint-based tutor that teaches EER modelling. EER-Tutor is a successor KERMIT, which was shown to significantly increase students' performance [11]. KERMIT is a stand-alone system and recently it has been re-implemented as a web-based tutoring system using WETAS, our web-based authoring shell [5]. For details of the system's architecture, please see [11]. Being a web-based application, EER-Tutor is divided into server and client modules. The server processes learners' solutions, generates feedback, and records all user actions. The client-side can be viewed in any common web browser as a set of dynamic HTML pages. The main page contains a Java applet shown in the centre of the browser window (Figure 1). The applet provides a set of drawing tools for creating EER diagrams. The navigation frame provides controls for stepping between problems, viewing session history and student model, opening EER-Tutor tutorial and help, printing current diagram and terminating the session. The frame at the bottom provides controls for submitting answers and viewing solutions to problems. Feedback is provided when the student submits the solution.

4. Re-engineering feedback for EER-Tutor

The specifics of EER modelling make it difficult to identify errors at the early stages of learning. A novice learner is in a vicious circle: trying to improve performance in some skill, the learner naturally does not intend to make errors but he/she is unable to detect errors, because of the lack of experience and knowledge. The same problem applies to the error correction stage: the learner must revisit faulty knowledge, but with open-ended tasks like EER modelling, the learner will have difficulty identifying relevant knowledge to correct.

CBM comes to the rescue. When an error occurs, the task of error detection and blame assignment is carried out by the system. The system should refer the learner to the relevant part of the domain knowledge. Consequently, an effective feedback message should tell the user (a) where exactly the error is, (b) what constitutes the error (perform blame allocation), and (c) refer the user to the underlying concept of the correct solution (revise underlying knowledge).

The above observations constitute the central focus of our experiment. Existing constraint-based tutors do not utilise these observations in feedback design. Feedback messages in EER-Tutor, as well as other tutors, merely tell the student to check a certain aspect of the solution and accompany a suggestion for correcting the problem. For example, consider the feedback message shown in Figure 1. The student has made a mistake when specifying a specialization of EMPLOYEE into MANAGER: this specialization should be partial (displayed as a single line in the diagram), while the students has specified a total specialization (double line). The intuitive feedback message that we have defined for this situation (coming from the violated constraint) is "*Check how you use subset connectors. In single subclass specialisations, subset connectors should be drawn with single lines.*" The student erroneously used a total specialization due to the lack of experience in extracting the modelling requirements from the problem statement. In particular, the phrase "*Some employees are managers*" in the problem text implies that the specialization should be partial. The error message partially allocates the blame and tells the student what has to be done to correct the error. However, this message does not point out the domain concept that the student has violated, and therefore does not offer help with the revision of underlying faulty knowledge. The message simply tells the learner what to do in order to correct the solution; this is insufficient for successful learning. On the contrary, the following message would (theoretically) have a greater impact: "*A* *specialisation with a single subclass is always partial (represented with a single line). Your solution contains a total specialisation with a single subclass".* This message starts with the general concept which caused the error, aimed at specialising the corresponding rule in the procedural memory, so that next time when a similar situation arises, the learner will hopefully be able to differentiate correctly between choosing partial or total participation. The second sentence ties the concept to the situation at hand, simultaneously pointing out the error and allocating the blame. The error correction information is not essential for the given problem, since there are only two options for specifying participation in a relationship. The careful engineering of every feedback message should theoretically influence learning. For the purposes of our study, we have redefined all feedback messages.

We suspect that common-sense feedback might result in *shallow learning*, which refers to failure in internalising the knowledge and poor knowledge transfer. In other words, the student might learn how to produce solutions that are correct from the system's points of view. However, the student would not be able to perform equally well in a different environment, as he/she does not really understand underlying domain concepts. This point is supported by research proving that learning how to play an educational game does not necessarily imply learning the target instructional domain [2]; learning happens only when students actively build the connections between his/her actions and underlying knowledge. In this light, we expect that the micro-engineered feedback in EER-Tutor will result in better knowledge transfer and deeper learning [12]. Another argument in support of the new feedback style originates from the ACT-R theory [1], but is equally applicable to CBM. The fourth principle of the ITSs design states that a tutoring system should promote an abstract understanding of problem-solving knowledge. This principle was motivated by the observation that students often develop overly specific knowledge from particular problem-solving examples; this is also related to shallow learning and poor knowledge transfer.

5. Evaluation

We performed a study at the University of Canterbury in August 2004. Second year students enrolled in an introductory database course were invited to participate. The students learned EER modelling concepts prior to the study during three weeks of lectures and had some practice during two weeks of tutorials. EER-Tutor was briefly introduced to the class in a lecture. The first session took place in a scheduled laboratory session. The participants were randomly allocated to one of the two versions of the system (referred to as control and experimental condition), differing only in the feedback style. The students were free to use EER-Tutor over two weeks. EER-Tutor contained 56 problems ordered in increasing order of difficulty. The students were not restricted in their choice of problems.

The first session started with an on-line pre-test and at the end of the two week period the students sat an on-line post-test. In this way, most students sat the pre-test in a supervised environment, but the post-test was offered to students in an uncontrolled environment. Two tests of comparable difficulty were interchangeably used for pre-and post-tests.

In order to maximise the effect of feedback, we introduced three restrictions to the users' interaction with the system. The system provided only one level of feedback, listing the messages of the first three errors at most. The students could not see the complete solution for the current problem unless they made at least five attempts at it. If the student saw the solution, the system would not allow further submissions for that problem.

105 students (82% of the class) participated in the study, the general statistics of which are given in Table 1. The maximum numbers of attempted and solved problems were 52 and 43 respectively, while interaction time ranged from 10 minutes to 45 hours. There are no significant differences between the two groups on all these measures. The difference between pre-test results is insignificant, indicating that the two groups had comparable prior knowledge.

Table 1. Statistics from the study

	Students	Time (hours)	Attempted problems	Solved problems	Feedback messages	Pre-test %	No Post-tests	Post-test %
Control	53	16.9 (12.6)	15.5 (11.4)	13.2 (10.3)	24.4 (22.1)	64.2 (26.7)	46	16.6 (7.3)
Exper.	52	15.9 (10.5)	15.2 (10.7)	12.9 (10.2)	23.5 (20)	59.6 (28.7)	45	26.5 (22)

As the post-test was administered on-line, not all students have submitted it, as reported in the table. The low post-test scores are due to many students not taking time to answer the questions. The log files show that many students submitted the post-test only a few seconds after the system displayed it. Even when the time between login and post-test submission is longer, we can not tell apart the situations when students did not answer questions at all or answered them incorrectly. The reason for this is that in the encoding scheme for the post-test results, both a no answer and an incorrect submission were recorded as zero. Consequently, we are unable to use the post-test results to compare the two groups.

We then analyzed how students learned constraints. If constraints represent appropriate units of domain knowledge, the learning should follow a smooth curve [1]. From the logs, we identified all relevant constraints for every attempt. Each constraint relevance occasion was rank-ordered from 1 up. We calculated, for each participant, the probability of violating each constraint at each attempt. The probabilities were then averaged across all the constraints all participants. The cut-off point is set at 50% of the initial number of relevant constraints. The resulting learning curves are shown in Figure 2.a. The probability of constraint violation for both groups decreases regularly (as evidenced by good fits to the power curves). The experimental group violated fewer constraints, and learned constraints faster: their learning rate (-0.2978) is higher than that of the control group (-0.2681).

Using the same approach, we calculated probabilities only for those constraints whose feedback messages had been seen by students, in order to focus on the effect of feedback. The resulting curves are shown in Figure 2.b. Power curve fits for the two groups are lower than in

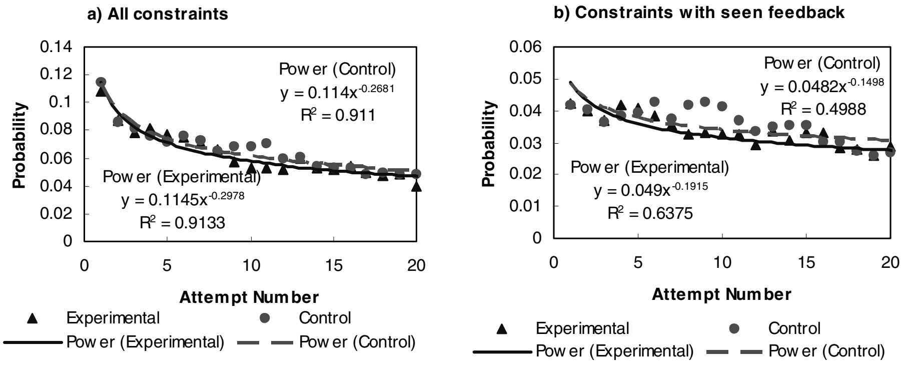


Fig. 2. Learning curves

Figure 2.a, but the probability is much lower, showing that students do learn from getting feedback. The learning rate is still higher for the experimental group.

We also analyzed the number of constraints learned as a function of the number of feedback messages received. If theory-based feedback is better than intuitive style, participants should acquire more knowledge, i.e. more constraints. From the logs, we identified for each participant the number of constraints they learned while interacting with the system. This analysis took into account only those constraints that were not known to the user at the start of the experiment. A constraint is considered as known by the student if the window of five attempts in the constraint history indicated successful application of this constraint in at least 80% cases. The number of learned constraints was then plotted as a function of received instruction, i.e. the number of seen feedback messages (Figure 3.a). The slopes of the trend lines indicate that experimental feedback resulted in more efficient constraint acquisition. Figure 3.b shows the results of the same analysis using a different criterion to test whether a constraint is learned. This time we used a window of three consecutive attempts, and considered a constraint as learned if it was used correctly two or three times.

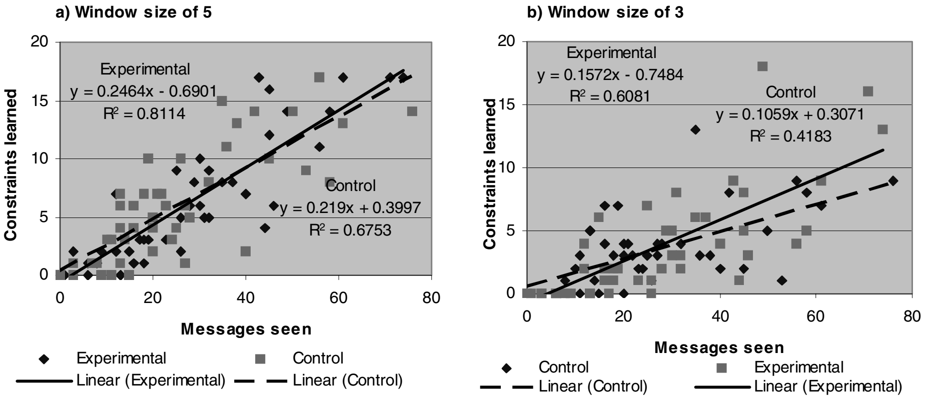


Fig. 3. Learned constraints as a function of feedback received

The participants used the system for a short time, and received a small number of feedback messages. The average number of messages received per one hour of instruction was 1.4, and the average number of learned constraints was 5.9 (sd=5.2). In a realistic situation, the system would be used for hundreds of hours. Using either figure, as participants spend more time with the system, and consequently get more feedback messages, the difference between the two styles of feedback becomes bigger. Extrapolating from Figure 3.a, after 140 feedback messages (based on 100 hours of learning), a student receiving old style feedback would have learned 31.1 constraints, while the theory-based feedback would result in 33.8 learned constraints.

5 Conclusions

This paper reported a project the goal of which was to investigate the role a learning theory might have in formulating feedback for intelligent tutoring systems. We noticed that guidelines for designing effective feedback are rare in research literature, which is strange given the fact that most educational systems claim to be based on various learning theories. Each theory proposes a view on learning, and therefore it should be possible to formulate to identify the principles of effective feedback based on the postulates of the chosen theory.

We have developed a number of constraint-based tutors, starting from the theory of learning from performance errors. This theory can be used by the ITSs to provide learners with extensive support during the learning process. A constraint-based tutor helps the student to identify an error in cases when the student does not have enough experience or knowledge to do that on their own. Effective feedback messages based on this theory should point out the error, and inform the student about the underlying domain principle that has been violated, thus making it possible for the student to revise faulty knowledge.

We hypothesized that principled, theory-based feedback should be more beneficial than intuitive feedback present in most existing systems, including our constraint-based systems. To test the hypothesis, we performed an experiment involving two versions of EER-Tutor, a system that teaches database design. The two version of the systems differed only in the style of feedback give to students. The study showed that feedback developed according to the learning theory provided better learning support, resulting in faster learning rates. The combination of the general domain knowledge relevant to the student's error, along with the specific details of the error in the given situation provides learning benefits through simultaneous revision of faulty knowledge and strategies.

This paper presented results of a study that lasted only two weeks. We plan to perform a longer study of similar nature. Furthermore, our results seem to have wider consequences, for educational systems based on other learning theories. We believe this is an interesting challenge for the whole IED community.

Acknowledgements: We thank Pramuditha Suraweera, Brent Martin and other members of ICTG for their invaluable help in developing EER-Tutor.

References

1. Anderson, J.R., Lebiere, C. The Atomic Components of Thought. Lawrence Erlbaum Associates, Mahwah, New Jersey, 1998.
2. Conati, C, Zhao, X. Building and Evaluating an Intelligent Pedagogical Agent to Improve the Effectiveness of an Educational Game. In Proc. 9th Int. Conf. on Intelligent User Interface, ACM Press, 2004, pp. 6–13.
3. Elmasri, R., Navathe, S. B. Fundamentals of Database Systems. Addison-Wesley, 2003, 4th edition.
4. Goel, V., Pirolli, P. Motivating the Notion of Generic Design with Information Processing Theory: the Design Problem Space. AI Magazine, v10, 1988, 19-36.
5. Martin, B., Mitrovic, A. Domain Modeling: Art or Science? In: U. Hoppe, F. Verdejo & J. Kay (ed) Proc. 11th Int. Conference on Artificial Intelligence in Education AIED 2003, IOS Press, pp. 183-190, 2003.
6. Mathan, S., Koedinger, K. An Empirical Assessment of Comprehension Fostering Features in an ITS. In: S. Cerri, G. Gouarderes and F. Paraguacu (eds) Proc. ITS 2002, Springer, 2002, pp. 330-343.
7. McKendree, J**.** Effective Feedback Content for Tutoring Complex Skills. Human-Computer Interaction, v5no4, 1990, 381-413.
8. Mitrovic, A., Ohlsson, S. Evaluation of a constraint-based tutor for a database language. Int. J. Artificial Intelligence in Education, v10 no3-4, 1999, 238-256.
9. Ohlsson, S. Constraint-Based Student Modelling. In J. Greer and G. McCalla (eds) Student Modelling: The Key to Individualised Knowledge-based Instruction, vol. 125 Computer Systems and Sciences, NATO ASI, Springer-Verlag, 1994, pp. 167–189.
10. Ohlsson, S. (1996) Learning from Performance Errors. *Psychological Review* **103**(2) 241-262.
11. Suraweera, P., Mitrovic, A. An Intelligent Tutoring System for Entity Relationship Modelling. Int. J. Artificial Intelligent in Education, v14no3-4, 2004, 375-417.
12. VanLehn, K., Freedman, R. et al. Fading and deepening: The next steps for ANDES and other model-tracing tutors. In G. Gauthier, C. Frasson, and K. VanLehn (eds) Proc. 5th Int. Conf. ITS 2000, Springer-Verlag, Montreal, 2000, pp. 474-483.

This page intentionally left blank

Posters

This page intentionally left blank

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

An Ontology of Situations, Interactions, Processes and Affordances to Support the Design of Intelligent Learning Environments

Fabio N. AKHRAS
Renato Archer Research Center
Rodovia Dom Pedro I, km 143,6
13069-901 Campinas, São Paulo, Brazil
E-mail: fabio.akhras@cenpra.gov.br

Abstract. In this paper we discuss the use of an ontology of situations, interactions, processes and affordances in the design of intelligent learning environments. Due to its broad scope, the ontology allows to provide an integrated view of the several aspects involved in the design of these systems. It also allows to address issues related to the connections between the context of learning and what is learned, between knowledge and activity, and between the process and the product of learning, which have been addressed by recent research on education.

Introduction

With the aim of making more precise the design of Intelligent Educational Systems (IESs), to facilitate the reuse and interoperability of the system's components, ontologies for AI in Education (AI-Ed) have addressed several aspects of these systems, such as learning tasks [1], communication between learners [2], learning goals [3], and group formation for collaborative learning [4], among others, in addition to the issue of how to integrate or interrelate ontologies [5].

In this paper, we present an ontology of situations, interactions, processes and affordances, and discuss its use in the design of Intelligent Learning Environments (ILEs). Due to its broad and integrated view of the various aspects involved in the design of ILEs, the ontology allows to address aspects that have been addressed by other ontologies, such as learning goals and learning tasks, in connection with other aspects that are relevant to the design of ILEs, such as the evaluation of learning and the adaptation of the learning environment. It also allows to address the connections between the context of learning and what is learned, between knowledge and activity, and between the process and the product of learning, which are issues that have been addressed by recent research on education.

1. ILE Design Based on the Ontology

Designing an ILE, according to the perspective provided by our ontology, involves four main steps: designing situations, designing interactions, designing processes, and designing affordances. These designs are structured within a framework that includes three main levels of abstraction, as shown in Figure 1. At the more abstract level there is the conceptual or axiomatic level provided by the definition of the ontology. At the more concrete level there is the instantial level produced by the particular occurrences of events, situations, patterns of interaction and properties of course of interaction, and by the particular affordances that hold in situation types as a consequence of these occurrences. At this instantial level the information produced constitute the models used to support the functions carried out by ILEs to evaluate learning and adapt the learning environment (the content of learner models, for example, would be part of this level).

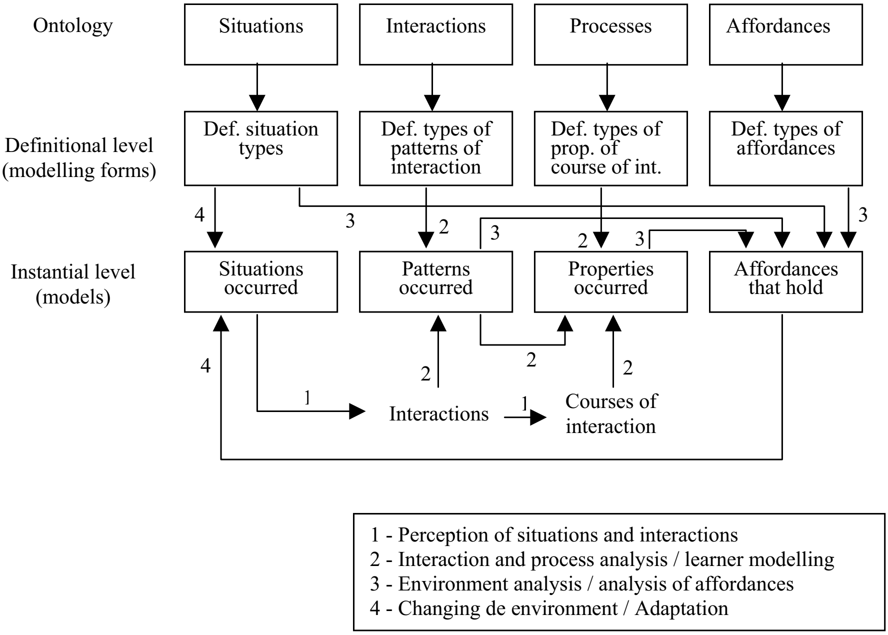

Ontology
Situations
Interactions
Processes
Affordances
Definitional level (modelling forms)
Def. situation types
Def. types of patterns of interaction
Def. types of prop. of course of int.
Def. types of affordances
Instantial level (models)
Situations occurred
Patterns occurred
Properties occurred
Affordances that hold
Interactions
Courses of interaction
1
2
3
4
1 - Perception of situations and interactions
2 - Interaction and process analysis / learner modelling
3 - Environment analysis / analysis of affordances
4 - Changing de environment / Adaptation

Figure 1. The structure of ILE design based on the ontology of situations, interactions, processes and affordances.

In the middle there is a definitional level in which the ontology is used to define modelling forms for issues that are relevant to the ILEs being designed. The modelling forms defined at this level will provide entities (and their formal definitions) which will become the types of the entities of the instantial level (for example, a definition of the types of entities that can occur in learner models would be part of this level). It will include forms of modelling interactions, in terms of particular patterns of interaction defined (e.g. utilises), which will be the types of patterns of interaction that can occur at the instantial level. Similarly, it will include forms of modelling processes, in terms of particular properties of course of interaction defined (e.g.: cumulative), which will be the types of properties of course of interaction that can occur at the instantial level. In addition, it will include forms of modelling affordances, in which the particular affordances defined will be the types of affordances that situation types can be expected to exhibit. Finally, the modelling of situation types, which will be the types of the situations that can occur at the instantial level, is also part of this level.

The issues modelled at the definitional level will address educational theories and AI-Ed approaches, according to the perspective of modelling situations, interactions, processes and affordances.

Therefore, particular views of learning being addressed will be reflected in the way situation types are designed, and in the types of patterns of interaction and of properties of course of interaction that will be relevant to consider in evaluating learning. In addition, the definition of the types of affordances that are relevant will be connected to the way it is intended that the system performs its role in supporting learning and adapting the learning environment, as it will determine the affordances to be considered in situation types in providing learning opportunities to the learners.

The models that are developed according to the ontology, at the levels of types and instances, integrate to support the main functions of ILEs. According to our framework, these functions, which are indicated by the numbers in Figure 1, correspond to:
(1) Perception of situations, events that occurred, and the changes they caused in situations.
(2) Reasoning to interpret these perceptions in terms of patterns of interaction and properties of course of interaction.
(3) Reasoning to infer, on the basis of the content and dynamics of the situation types available, the affordances that hold in those situation types.
(4) Decision making to change the context of learning, providing situation types with the kinds of affordances that can enable the achievement of learning goals.

2. Conclusion

In this paper we have presented a framework to support ILE design based on an ontology of situations, interactions, processes and affordances.

We have used our ontology to develop modelling forms that operationalize issues of constructivist theories of learning, and have developed general approaches to support knowledge representation, reasoning and decision making in ILEs based on the models that can be developed based on the modelling forms. These approaches and models have been implemented in INCENSE [6], an ILE in the domain of software engineering. This system has been implemented and exposed to use, and showed that the ontology can be used to support the various aspects of ILE design, including the definition of contexts for learning, the definition of patterns of interaction and properties of course of interaction, used to evaluate learning, and the definition of affordances, used to guide the adaptation of the learning contexts, in a way that has allowed the ILE to evaluate learning and adapt the learning environment in constructivist terms.

Acknowledgements. This work has been partially financed by CNPq, Brazil

References

[1] Mizoguchi, R.; Sinitsa, K. and Ikeda, M. (1996) Knowledge engineering of educational systems for authoring systems design. Proceedings of the European Conference on Artificial Intelligence in Education (EuroAIED), pp.329-335.

[2] Ikeda, M.; Hoppe, H. U. and Mizoguchi, R. (1995). Ontological issues of CSCL systems. Proceedings of the 7th International Conference on Artificial Intelligence in Education (AIED´95), pp. 242-249.

[3] Inaba, A.; Supnithi, T.; Ikeda, M.; Mizoguchi, R. and Toyoda J. (2000). An Overview of "Learning Goal Ontology". Proc. ECAI'2000 Workshop on Analysis and Modelling of Collaborative Learning Interactions.

[4] Supnithi, T.; Inaba, A.; Ikeda, M.; Toyoda, J. and Mizoguchi, R. (1999). Learning Goal Ontology Supported by LearningTheories for Opportunistic Group Formation. Proceedings of the 9th International Conference on Artificial Intelligence in Education (AIED´99), pp 67-74, Le Mans France.

[5] Barros, B.; Verdejo, M. F.; Read, T. and Mizoguchi, R. (2002). Applications of a Collaborative Learning Ontology. MICAI'2002 Mexican International Conference on Artificial Inteligence. LNAI, Springer-Verlag, pp.301-310.

[6] Akhras, F. N. and Self, J. A. (2000). System intelligence in constructivist learning. *International Journal of Artificial Intelligence in Education,* 11(4):344-376.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Toward supporting hypothesis formation and testing in an interpretive domain

Vincent Aleven[1] and Kevin Ashley[2]

[1]*Human-Computer Interaction Institute*
Carnegie Mellon University

[2]*Intelligent Systems Program, Learning Research and Development Center*
University of Pittsburgh

Abstract. The research field of AI & Education has long been interested in cognitive processes in which students formulate and test hypotheses by considering them in light of specific cases. However, few if any of the systems that have been built target domains which are ill-structured and in which determining whether a hypothesized rule and proposed outcome are consistent with past decisions is a matter of interpretation, rather than deductive inference. The goals of our project are to (1) develop an AI model of hypothesis formation and testing in an interpretive domain, US Supreme Court oral arguments and (2) to use it in an intelligent tutoring system to guide law students in learning that process. As a first step toward these goals we will conduct an experiment to evaluate whether self-explanation prompts facilitate learning by studying argument transcripts.

Introduction

The research field of AI & Education has long been interested in processes of inquiry learning in which students formulate hypotheses and test them against specific cases ([1-5]). These processes as they occur in ill-structured domains, however, have received little attention, one exception being the work on CATO [6-8]. Oral arguments before the United States Supreme Court offer prime examples of hypothesis formulation and testing in an ill-structured domain. In these arguments, advocates frame hypotheses for deciding a case and the Justices challenge them, often by posing hypothetical scenarios that test the hypotheses' limits. While these processes in the legal domain bear some resemblance to the corresponding processes in science or mathematics, in the legal domain determining whether a hypothesized rule and proposed outcome are consistent with past decisions and plausible hypotheticals is much more a matter of interpretation.

The goals of our project are (1) to develop a computational model of the reasoning processes exemplified in US Supreme Court oral argument and (2) to use the model as the basis for an intelligent tutoring system that will engage students in an appropriately simplified version of these processes.

A planned experiment

As a first step, we will run an experiment to find out whether specific prompts for self-explanation help students gain a deeper understanding, as they study transcripts of Supreme Court oral arguments. The cognitive science literature supports that studying examples is an effective learning strategy at the early stages of acquiring a cognitive skill [9] and that self-explanation prompts can help students gain a deeper understanding of the subject matter ([10-12]). However, the effectiveness of prompts has not yet been shown in ill-structured domains as complex as the legal reasoning exemplified in Supreme Court oral

Table 1: Excerpt of transcript of oral argument made before the US Supreme Court in *California v. Carney,* 105 S. Ct. 2066 (1985), with self-explanation prompts added

Argument Transcript	Self-Explanation Prompts
QUESTION: Well, what if the vehicle is in one of these mobile home parks and hooked up to water and electricity but still has its wheels on?	
MR. HANOIAN: [*9] If it still has its wheels and it still has its engine, it is capable of movement and it is capable of movement very quickly.	1. Do you think H's response is effective?
QUESTION: Even though the people are living in it as a home and are paying rent for the trailer space, and so forth?	2. Why are the Justices adding these features to the hypothetical?
QUESTION: Well, there are places where people can plug into water, and electricity, and do. There are many places, for example, in the state I came from where people go and spend the winter in a mobile home. And you think there would be no expectation of privacy in such circumstances?	3. Why does it matter whether there would be expectations of privacy? 4. If it was clear that there is, or should be, a high expectation of privacy in the current fact situation, would that favor H's position? 5. Nothing is said from which we can infer how this particular hypothetical should be decided. Does that matter? That is, what good is it to use hypotheticals whose outcome is unknown? Wouldn't it be better to cite past cases, whose outcome we do know?
MR. HANOIAN: Well, I am not suggesting that there is no expectation of privacy in those circumstances, Your Honor.	6. By conceding that there are expectations of privacy in the hypothetical scenarios sketched by the judges, does H not reduce his chances of winning the case at hand? 7. Does H concede that the mobile home park hypothetical should have the opposite result as the case at hand? 8. How would H distinguish the current case from the mobile home park hypothetical?

arguments. In light of the evidence that prompts do not benefit all students equally ([10, 11]), it is important to ask how effective prompts are in such challenging domains.

Table 1 shows excerpts from oral arguments made in the case of *California v. Carney,* 105 S. Ct. 2066 (1985), with self-explanation prompts inserted. This case involved the legality under the 4th Amendment of the US Constitution of a warrantless search of a motor home located in a downtown San Diego parking lot. Police suspected defendant Carney of trading marijuana for sex acts. After they questioned a boy leaving Carney's motor home, agents entered the motor home without a warrant or Carney's consent, observed marijuana, and arrested Carney. The case pitted two conflicting principles: the State's right to deal effectively with the exigent possibility that evidence of a crime will disappear versus the citizen's constitutionally protected expectation of autonomy and privacy in his home. In the oral argument, the State's attorney, Mr. Hanoian, proposed a bright line test: if the vehicle/home is capable of self-locomotion, then no warrant is required to search it. As shown in Table 1, he then has to respond to the Justice's challenge hypothetical: what result would his test produce when applied to a summer motor home with wheels that is hooked up to utilities? Mr. Hanoian responds that such a vehicle still might be moved in a hurry, but concedes the owners would have some expectation of privacy. Some of the self-explanation prompts focus on the effectiveness of that response. Others focus on the Justices' strategies and possible reasons for posing hypotheticals.

Discussion

In order to evaluate the effect of the self-explanation prompts, the study will compare the learning results of students studying argument transcripts with and without self-explanation prompts. A pilot study involving two law students, a first-year student and a second-year student, provided some evidence that the prompts are useful. The students went through the *Carney* transcript twice, the first time without self-explanation prompts, the second time with. Each time, they were asked to answer a number of questions about the argument exchange they had just studied. We saw a difference in the quality of the answers between the first-year and the second-year student, indicating that the material is challenging. Further, we saw that the answers of the first-year student improved, after studying the transcript with the self-explanation prompts. Of course, such evidence is preliminary, due to the "small N". Also, the improvement in the answers could be attributed simply to the fact that the student went through the transcript twice. This confound will be avoided in the actual experiment by having a control group. We are currently working on developing a suitable task by which we can measure any improvement in students' argument-making capabilities, a preliminary challenge for any research in an ill-structured domain.

We expect the study to yield information about how students understand and make arguments. This information will help us start to build an argument model and develop an intelligent tutoring system. The study will also contribute to cognitive science by testing whether specific self-explanation prompts can help students to learn to engage in a process of hypothesis formation and testing in an ill-structured domain.

Acknowledgements

This research is sponsored by NSF Award IIS-0412830. The contents of the paper are solely the responsibility of the authors and do not necessarily represent the official views of the NSF.

References

[1]. Bunt, A., C. Conati, and K. Muldner, Scaffolding Self-Explanation to Improve Learning in Exploratory Learning Environments, in *Proceedings of the 7th International Conference on Intelligent Tutoring Systems, ITS 2004,* J. Lester, R.M. Vicario, and F. Paraguaçu, Editors. 2004, Springer: Berlin.

[2]. Collins, A. and A.L. Stevens, Goals and Strategies of Inquiry Teachers, in *Advances in Instructional Psychology,* R. Glaser, Editor. 1982, Lawrence Erlbaum: Hillsdale, NJ. p. 65-119.

[3]. Murray, T., L. Winship, and N. Stillings, Evaluation of the SimForest Inquiry Learning Environment: Inquiry Cycles and Collaborative Teaching Practices, in *AERA.* 2004: San Diego, CA.

[4]. Shute, V.J. and R. Glaser, A Large-Scale Evaluation of an Intelligent Discovery World: Smithtown. *Interactive Learning Environments,* 1990. 1: p. 51-77.

[5]. Woolf, B.P., et al., Tracking Student Propositions in an Inquiry System, in *Proceedings of the 11th International Conference on Artificial Intelligence in Education, AIED 2003,* U. Hoppe, F. Verdejo, and J. Kay, Editors. 2003, IOS Press: Amsterdam. p. 21-28.

[6]. Aleven, V., Using Background Knowledge in Case-Based Legal Reasoning: A Computationl Model and an Intelligent Learning Environment. *Artificial Intelligence,* 2003. 150: p. 183-238.

[7]. Aleven, V. and K.D. Ashley, Teaching Case-Based Argumentation Through a Model and Examples: Empirical Evaluation of an Intelligent Learning Environment, in *Proceedings of the 8th International Conference on Artificial Intelligence and Education, AI-ED '97,* B. du Boulay and R. Mizoguchi, Editors. 1997, IOS Press: Amsterdam. p. 87-94.

[8]. Ashley, K.D., R. Desai, and J. Levine, Teaching Case-Based Argumentation Concepts Using Dialectic Arguments vs. Didactic Explanations, in *Proceedings of the Sixth International Conference on Intelligent Tutoring Systems, ITS 2002,* S.A. Cerri, G. Gouardères, and F. Paraguaçu, Editors. 2002, Spinger: Berlin. p. 585-595.

[9]. Atkinson, R.K., et al., Learning from examples: Instructional principles from the worked examples research. *Review of Educational Research,* 2000. 70(2): p. 181-214.

[10]. Chi, M.T.H., et al., Eliciting Self-Explanations Improves Understanding. *Cognitive Science,* 1994. 18: p. 439-477.

[11]. Renkl, A., et al., Learning from Worked-Out Examples: the Effects of Example Variability and Elicited Self-Explanations. *Contemporary Educational Psychology,* 1998. 23: p. 90-108.

[12]. Schworm, S. and A. Renkl, Learning by solved example problems: Instructional explanations reduce self-explanation activity, in *Proceeding of the 24th Annual Conference of the Cognitive Science Society,* W.D. Gray and C.D. Schunn, Editors. 2002, Lawrence Erlbaum: Mahwah, NJ. p. 816-821.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Authoring plug-in tutor agents by demonstration: Rapid, rapid tutor development

Vincent Aleven and Carolyn Rosé
Human-Computer Interaction Institute
Carnegie Mellon University
Pittsburgh, PA

Abstract We combined two existing methods for rapid tutor development: "plug-in tutor agents" [6] and an authoring tool suite (CTAT) that supports the creation of tutors "by demonstration" [2]. The combined approach, which has not been tried before, is suited for adding tutoring capabilities to an existing problem-solving environment, for example an off-the-shelf simulator. Connecting the components (i.e., the simulator and CTAT) requires programming but once that is done, "Pseudo Tutors" are created "by demonstration. Following this approach, we created plug-in Pseudo Tutor agents for a thermodynamics simulator, CyclePad [1], which were tried out in a classroom experiment involving 92 college students. The experiment demonstrates that the Pseudo Tutor technology is viable in a complex domain and that Ritter and Koedinger's protocol for the tool-tutor communication is suited for use in an authoring environment.

Introduction

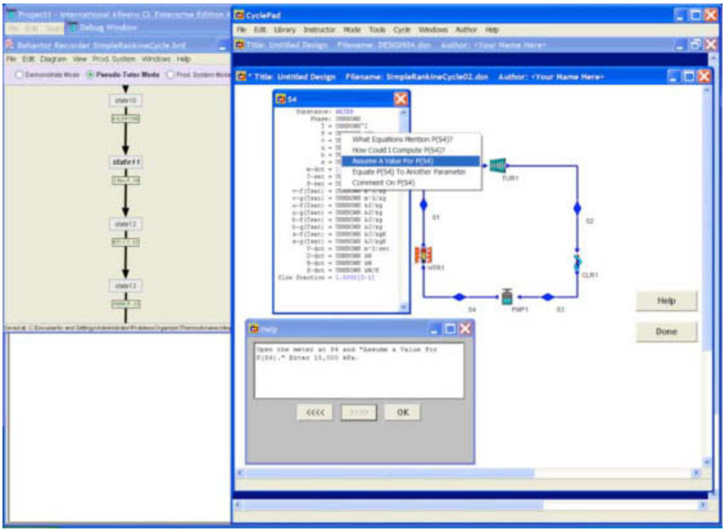
It has long been recognized that it takes much time and effort to build an intelligent tutoring system. A number of approaches have been tried to bring down the development time and to make ITSs easier to develop. An approach that holds much promise is the use of "Plug-In Tutor Agents," [6], a way of adding tutoring to existing problem-solving environments or simulators (referred to as "tool") without having to build a complete tutor from scratch. Examples of this approach are the science learning space [3] and the Excel tutor [4]. A key contribution of this work is the specification of a protocol for the tool-tutor communication.

A different approach to rapid tutor development is the creation of authoring tools, which typically facilitate the development of the knowledge sources for an ITS. A wide array of tools have been developed and some have proven capable of supporting the development of effective tutoring systems [5]. The Cognitive Tutor Authoring Tools (CTAT) support the development of so-called Pseudo Tutors, which can be created without programming, namely, by demonstrating correct and incorrect solutions to tutor problems [2]. Pseudo Tutors are problem-specific, but in many domains they offer an attractive trade-off between development time and generality. Preliminary investigations indicate that it takes considerably less time to develop Pseudo Tutors than full-blown Cognitive Tutors [2].

Since authoring tools and plug-in tutor agents have complementary advantages, it seems natural to combine them. This poster describes the use of CTAT to add tutoring to the CyclePad thermodynamics simulator [1]. This AI-based system, shown on the right in Figure 1, lets students build and analyze thermodynamic "cycles" such as those underlying power plants, combustion engines, and refrigerators..

Developing Pseudo Tutors for CyclePad

To add tutoring to CyclePad, we hooked it up to CTAT (see also [6]): First, we made the simulator "recordable," so that it communicates all actions that the user takes in the simulator to CTAT, (more or less) conform to the protocol for tool-tutor communication laid out in [6]. This information enables the Pseudo Tutor to track students as they work through problem scenarios. This step was not especially difficult.


Debug Window
Behavior Recorder SimpleRankineCycle.brd
File Edit Diagram View Prod. System Windows Help
○ Demonstrate Mode ● Pseudo Tutor Mode ○ Prod. System Mode
state10
state11
state12
state13
CyclePad
File Edit Library Instructor Mode Tools Cycle Windows Author Help
Title: Untitled Design Filename: DESIGN34.dsn Author: <Your Name Here>
* Title: Untitled Design Filename: SimpleRankineCycle02.dsn Author: <Your Name Here>
S4
What Equations Mention P(S4)?
How Could I Compute P(S4)?
Assume A Value For P(S4)
Equate P(S4) To Another Parameter
Comment On P(S4)
TUR1
S1
S2
HTR1
CLR1
S4
PMP1
S3
Help
Done
Help
<<<<
>>>>
OK

Open the meter at S4 and "Assume a Value for P(S4)." Enter 13,000 kPa.

Figure 1: Using CTAT's Behacvior Recorder (left) to develop plug-in Pseudo Tutors for CyclePad (right)

Second, we made the simulator "scriptable," that is, capable of responding to scripting commands from CTAT. Scriptability is necessary so that the tutor can prevent incorrect student actions from being executed and prevent the student from undoing the effect of correct actions that have already been OK-ed by the tutor. Without this ability, the tutor may loose track of where the student is in the given problem scenario, which would lead to ineffective tutoring. It has been suggested [6] that the responsibility for undoing incorrect actions may be placed in the hands of the student, obviating the need for this kind of scriptability. However, we do not see that as a viable option for real-world tutors. Since CyclePad did not come with an undo facility, we implemented one ourselves, which required a fair amount of effort. The communication from tutor to tool roughly conformed to the protocol specified in [6].

Finally, we made the tutor visible in the CyclePad interface. This amounted mainly to adding a Help button and a messages window, in which CyclePad displays the tutor's hint and error feedback messages. With CyclePad hooked up to CTAT, we created Pseudo Tutors for scenarios in which students create, analyze, and optimize three different designs for the so-called Rankine cycle, a key thermodynamic design that is the basis for the steam-based power plants that generate the majority of the electricity in the US. These Pseudo Tutors were created in the usual manner, without programming: by demonstrating correct and incorrect steps, and annotating them with hints and feedback messages [2].

A classroom evaluation of the thermodynamics Pseudo Tutors

The thermodynamics Pseudo Tutor plug-ins were evaluated in a controlled experiment, carried out in an undergraduate thermodynamics course. The experiment (described more fully in [7]) involved 92 students, mostly sophomores, of whom 39 used the Pseudo Tutors. The goal of the experiment was to compare students' learning results in three different conditions in which they used the CyclePad simulator guided by, respectively, a script (on paper), Pseudo Tutor plug ins, and a human tutor in a Wizard of Oz scenario. There were reliable learning gains in all conditions including the Pseudo Tutor condition. The human tutor was reliably better than the other two conditions; the difference between the other two conditions was not reliable (t-test, $p > 0.2$). The Pseudo Tutor condition was plagued by a number of technical problems, none of which had to do with CTAT. Also, due to the severe time pressure under which we had to prepare for the experiment, we managed to provide Pseudo Tutors for only 2 of the 3 scenarios that the students explored with CyclePad. For these reasons, we do not consider the experiment to have been a fair test of the Pseudo Tutor technology. It did however constitute the first use of Pseudo Tutors in a university classroom and did show learning gains associated with the use of Pseudo Tutors.

Discussion and conclusion

By combining two approaches to rapid tutor development, plug-in tutor agents and authoring tools, we were able, in a short period of time, to add tutoring to CyclePad, a sophisticated simulation environment. Establishing the required communication between CyclePad and CTAT was not trivial, but neither did it require very extensive development effort. Most of the effort went into implementing an undo facility in CyclePad. The communication between CyclePad and CTAT was based roughly on the protocol presented in [6], which, originally developed to support tutoring, was by and large capable of supporting authoring as well.

Once CyclePad was hooked up, we developed tutoring capabilities without programming. A classroom experiment showed that Pseudo Tutors are a viable way to provide tutoring in a domain as complex as thermodynamics, even though, due to a number of technical problems, the gains were not as large as we had hoped. Given the time constraints under which the Pseudo Tutors were developed, we feel confident that we will be able to improve the Pseudo Tutors significantly. The experiment was to the best of our knowledge the first foray into a college classroom of a system based on the plug-in tutor principle. Further, while Pseudo Tutors have before been used before for high-school math, in a genetics college course, and in a number of on-line courses at Carnegie Mellon, the Pseudo Tutors for thermodynamics were the first controlled evaluation this technology in a college classroom. For those wanting to try out the development approach described here, the Cognitive Tutor Authoring Tools are available free of charge, for research and educational purposes, at http://ctat.pact.cs.cmu.edu.

Acknowledgments

This research is sponsored by ONR Award N000140410107 and an NSF Award SBE0354420 to the Pittsburgh Sciences of Learning Center. The contents of the paper are solely the responsibility of the authors and do not necessarily represent the official views of the ONR or the NSF. We thank Chang-Hsin Chang, Jonathan Sewall. Zhenhua Zhang, Chih-Yuh Chao, Christine Wilson, and Sandy Katz for their contributions.

References

[1] Forbus, K. D., Whalley, P. B., Evrett, J. O., Ureel, L., Brokowski, M., Baher, J., Kuehne, S. E. (1999). CyclePad: An Articulate Virtual Laboratory for Engineering Thermodynamics. *Artificial Intelligence 114*(1-2): 297-347.

[2] Koedinger, K. R., Aleven, V., Heffernan, N., McLaren, B., & Hockenberry, M. (2004). Opening the Door to Non-Programmers: Authoring Intelligent Tutor Behavior by Demonstration. In J. C. Lester, R. M. Vicario, & F. Paraguaçu (Eds.), *Proceedings of Seventh International Conference on Intelligent Tutoring Systems, ITS 2004* (pp. 162-174). Berlin: Springer Verlag.

[3] Koedinger, K. R., Suthers, D. D., & Forbus, K. D. (1999). Component-based construction of a science learning space. *International Journal of Artificial Intelligence in Education, 10.*

[4] Mathan, S., & Koedinger, K. R. (2003). Recasting the feedback debate: Benefits of tutoring error detecting and correction skill. In U. Hoppe, F. Verdejo, & J. Kay (Eds.), *Proceedings of the 11th International Conference on Artificial Intelligence in Education, AI-ED 2003.* Amsterdam: IOS Press.

[5] Murray, T., Blessing, S., & Ainsworth, S. (Eds.) (2003). *Authoring Tools for Advanced Technology Learning Environments: Towards cost-effective adaptive, interactive and intelligent educational software.* Dordrecht, The Netherlands: Kluwer.

[6] Ritter, S., & Koedinger, K. R. (1996). An architecture for plug-in tutor agents. *Journal of Artificial Intelligence and Education 7*(3/4), 315-347.

[7] Rosé, C. P., Aleven, V., Carey, R., & Robinson, A. (in press). A First Evaluation of the Instructional Value of Negotiable Problem-Solving Goals on the Exploratory Learning Continuum. *Proceedings of 12th International Conference on AI in Education, AIED 2005.*

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Evaluating Scientific Abstracts with a Genre-specific Rubric

Sandra ALUÍSIO[1], Ethel SCHUSTER[2], Valéria FELTRIM[1], Adalberto PESSOA Jr.[3], Osvaldo OLIVEIRA Jr.[4]
[1]*NILC/ICMC, University of São Paulo, Brazil, {sandra, vfeltrim}@icmc.usp.br*
[2]*Northern Essex Community College, USA, schuster@cs.uml.edu*
[3]*FCF, University of São Paulo, Brazil, pessoajr@usp.br*
[4]*NILC/IFSC, University of São Paulo, Brazil, chu@if.sc.usp.br*

Abstract. We propose a rubric to analyze abstracts of scientific papers. It is based on experiments carried out with a writing tool to assist non-native English speakers in producing abstracts in the field of Pharmaceutical Sciences, referred to as SciPo-Farmácia. The rubric has a 3-fold purpose: (1) as a facility to be plugged into SciPo-Farmácia to score students texts, (2) to help domain experts improve both the contents and the language of the text; and (3) to evaluate published articles in order to extend the text database of SciPo-Farmácia .

Introduction

The writing of scientific papers poses major difficulties for non-native English authors not only at the lexical and syntactic levels, but mostly at the textual level. Many problems include a) the use of rhetorical structures transferred from their mother language; b) misuse of logical relations, and c) lack of references and conventional expressions akin to the scientific discourse. To employ adequate rhetorical strategies, the writer needs to master the use of expressions and logical connections that are common to the scientific discourse. Non-native English writers can find help mainly for post-writing evaluation. One difficulty remains though, namely in *the production of a first draft that captures the essence of the work and that can be later improved with changes*. The errors and omissions mentioned above affect the coherence of the text, and this can only be tackled with tools that go beyond the post-writing evaluation. One option is to use a tool that provides linguistic material from published papers – in context and indexed - to facilitate its quick access. The reuse of linguistic material has been the core of AMADEUS (AMiable Article DEvelopment for User Support), a system for assisting non-native English speakers at various levels of expertise in their scientific writing [1, 2, 3, 4]. Based on it, we developed SciPo [5], a Web critiquing system for academic writing of theses in Computer Science, in Portuguese and SciPo-Farmácia (http://www.nilc.icmc.usp.br/SciPo-farmacia/), a version used in an English technical writing course in the Faculty of Pharmaceutical Sciences at University of São Paulo (USP), Brazil. This paper introduces a new rubric developed from experiments with SciPo-Farmácia. Although it currently handles English abstracts, the rubric will be extended to focus on other parts of scientific papers. In addition, part of the work done with the rubric will be automated. To our knowledge, there are no available systems to score scientific texts, although there are systems which score more general purpose texts [6, 7].

1. Experiments with SciPo-Farmácia

We carried out experiments with 7 graduate students enrolled in an English (scientific) writing course in order to identify the characteristics of their writing that should be covered by the rubric. The experiments consisted of two tasks. In the first task, the students were asked to write an abstract about their work in English. The abstract had to be done from scratch with no help. After this task, the students were trained to use SciPo-Farmácia. In the second task, the students were asked to do the same tasks as before but now with the help of SciPo-Farmácia only (and no other type of assistance). The abstracts were evaluated by three experts, in the following order. First, a scientific writing researcher checked for the presence of discourse components, its organization and balance. Next, a computational linguist assessed grammar and language usage. Finally, a domain expert evaluated the texts for their content and use of technical terms. After the first task, we observed that 4 abstracts had structural problems, such as lack of main components (4 of them) and ordering problems (one of them). After using the SciPo-Farmácia tool in the second task, the structure of two of the students' abstracts improved. For the remaining students, no significant changes were observed, even though they all stated that they were better prepared to organize their abstracts. The overall content of the abstracts, however, did not improve with the use of SciPo-Farmácia, mainly because these students lacked sufficient knowledge of English. Indeed, several errors – classified as global and local – were encountered in the evaluation. We list the global errors in dimension 5. Thus, even though the rubric to be created is aimed at assessing skills in scientific writing, it must contemplate the problems related to the language, particularly the global errors which may affect comprehension.

2. Rubric Development

Based on the error analysis in the experiments just described we developed a scoring rubric (Figure 1) aimed both at assessing skills in scientific writing and at handling the specific problems related to the language. It includes seven dimensions with two scale values each: high and low. Having only two scale values helps in both annotating dimensions and achieving high consistency among the human judges. Dimensions 1 and 2 give a rank of the abstract as a whole while the remaining rank individual sentences. Some dimensions are applied only to certain components.

Dimension 1: Characterization, Organization and Development. Deals with the presence of main components and their ordering in the abstract. While not all the abstracts follow a strict order presented in the model provided by the writing tool (Background, Gap, Purpose, Methodology, Main Results and Conclusion), some local ordering must be respected in order to facilitate its reading. **High** score if: a) the main components are present and follow the proper order: Purpose, Methodology (if any), Main Results, and Conclusion; b) if there is a Gap, it must be followed by the Purpose; c) if there is a Background and a Gap, the latter must follow the former. Also, it is possible to have cycles of Background and Gap. Otherwise, the score is **low**.

Dimension 2: Balance among the Components. Abstracts should not exceed 300 words. We found that most published abstracts in the field of Pharmaceutical Sciences have less than 30% of the total number of words dedicated to the background. In addition, the Purpose component is expected to be contained in one sentence**.** **High** score if: a) the Purpose exists and is written in one sentence; b) the Conclusion exists and is written in one sentence; c) If there is a Background, it should not exceed 30% of the words in the abstract. Otherwise, the score is **low**.

Dimension 3: Coherence among components. The components should be semantically related to each other, thus contributing to the coherence of the text. We have identified three relations among them: 1) fulfilment, that holds between Purpose and Gap, 2) accomplishment, which holds between Results and Purpose, and 3) generalization holds between Conclusion and Results. **High** score if a) the Purpose is related to the Gap in a fulfilment relation (if there is any Gap at all as this component is not required, otherwise the Purpose's label is assigned as default (N/A)); b) the Main Results is related to the Purpose in an accomplishment relation; c) the Conclusion is related to the Main Results in a generalization relation. If these relations do not hold, the first component is ranked **low** and the second one is marked as **default** (N/A).

Dimension 4: Cohesive markers. The sentences within each component must be cohesive. This is marked by discourse markers, pronominal reference or noun re-introduction. High score if each sentence is related to at least one other sentence in the component, otherwise the sentence receives a **low** rank. If the discourse component contains only one sentence, then dimension 4 is assigned a **default** (N/A) value. If there are cycles of Background (B) and Gap (G), then both components are evaluated as one for dimension 4.

Dimension 5: Technical errors. Here we consider only global errors. Frequent patterns of global errors (one of them, at minimum) are ranked with a **low** score. They include: transfer from mother tongue; word by word translation; sentence fragments; inappropriate choice of verb tense for specific component; lack of referent; word order error; misreading caused by missing the word "that"; missing discourse markers. A **high** score represents no technical errors.
Dimension 6: Style. The score is **high** if it contains no personal or colloquial style: e.g. I, my, me, frankly, by the way; emphatics (e.g. a lot, for sure, really); discourse particles (e.g. sentence-initial well, now, anyway); hedges such as I mean, I think, I assume, sort of, kind of, you know. Otherwise the score is **low**.
Dimension 7: Factual information. We recommend that authors produce informative (or descriptive) content though they may prefer to present indicative content. The score is **high** if it provides informative or substantive material in Main Results and Conclusion sentences without requiring further reference to the entire paper. The dimension 7 label for the remaining sentences is assigned as **default** (N/A).

Figure 1. 7-dimension rubric to evaluate scientific abstracts

Each sentence must be annotated according to the components of the rhetorical structure. In SciPo-Farmácia we used: Background, Gap, Purpose, Methodology, Main Results and Conclusion. These categories will be assigned manually to implement the discourse components classifier and will be generated automatically by a discourse analysis classifier in the implementation as a writing facility. We have already implemented a classifier to automatically annotate an abstract in Portuguese with a similar structure [5], which will be adapted for English.

3. Conclusion

The need for an unambiguous rubric that provides clear guidelines with a 3-fold purpose has been shown: for domain experts to evaluate scientific abstracts, for students to make revisions on their texts and for system developers to tag corpora with the main discourse components. Students who must write scientific abstracts can benefit significantly from the following. (1) Practice, by reading abstracts in their field in English. This will enable them to feel comfortable with the language itself. SciPo-Farmácia can facilitate this process by providing multiple examples that the students can look at for illustration. (2) Computer-based tools: Our work has shown that the use of SciPo-Farmácia tools can enhance students' level of confidence and thus enable them to improve the structure of their abstracts. (3) Formal evaluation techniques provided by a strict rubric: this can provide a measure for the students (as to how well or how poorly the abstract is written), for the evaluator (to provide a standard that measures all the abstracts in the same manner) and for a teacher whose goal is to improve the writing of his/her students.

References

[1] Aluísio, S.M. & Oliveira Jr. O.N. A case-based approach for developing writing tools aimed at non-native English users. Lectures Notes in Artificial Intelligence 1010, 121-132 (1995).

[2] Aluísio, S.M. & Gantenbein, R.E. Towards the application of systemic functional linguistics in writing tools. Proceedings of International Conference on Computers and their Applications, 181-185 (1997).

[3] Aluísio, S.M., Barcelos, I., Sampaio, J. and Oliveira Jr., O. How to learn the many unwritten "Rules of the Game" of the Academic Discourse: A hybrid Approach based on Critiques and Cases. In Proceedings of the IEEE International Conference on Advanced Learning Technologies. 257-260 (2001).

[4] Aluísio, Sandra M.; Aquino, Valéria T.; Pizzirani, Rafael; Oliveira Jr, Osvaldo Novais de. High Order Skills with Partial Knowledge Evaluation: Lessons learned from using a Computer-based Proficiency Test of English for Academic Purposes. Journal of Information Technology Education, v. 2, n. 1, 185-201 (2003).

[5] Feltrim, Valéria; Pelizzoni, Jorge Marques; Teufel, Simone; Nunes, Maria das Graças Volpe; Aluísio, Sandra M. Applying Argumentative Zoning in an Automatic Critiquer of Academic Writing. In: Proceedings of 17th Brazilian Symposium on Artificial Intelligence, v. 1, 214-223 (2004).

[6] Kukich, K. Beyond Automated Essay Scoring. IEEE Intelligent Systems, 15(5), 22-27 (2000).

[7] Landauer, T.K., & Laham, D. The Intelligent Essay Assessor. IEEE Intelligent Systems, 15(5), 27-31 (2000).

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Dynamic Authoring in on-line Adaptive Learning Environments

A. ALVAREZ, I. FERNÁNDEZ-CASTRO AND M. URRETAVIZCAYA
Department of Languages and Computer Systems
University of the Basque Country Apdo. 649, 20080 Donostia, Spain
e-mail.{ainhoa.alvarez, isabelfc}@ehu.es

Abstract. This paper is centred on the issue of adaptivity in the context of distributed learning systems. It presents an approach to improve the adaptivity aspect through a multi-session learning process. It is based on the concept of *dynamic generation of the tutoring knowledge* according to the changing characteristic emerging for each session.

This approach has produced the MAGADI platform. It is inspired in the structure of IRIS[2] and the architecture of the generated tutors (INTZIRI) has evolved to a distributed multi-agent system. Here its architectural development and authoring aspects are described.

Introduction

One of the main actual tendencies on educational environments is the development of Intelligent Learning Management Systems (ILMS). These try to incorporate adaptation mechanisms coming from the ITS field to LMS, but the difficulties for including this adaptation aspect have impeded to obtain the expected success [5].

Authoring tools provide a good start point for defining, and translating to LMS adaptation issues. Murray [4] identifies some desirable features for authoring tools, i.e. rapid prototyping, flexible design, content modularity, re-usability, customisation and extensibility. However, one main critique is that the tutors they generate use to maintain a static adaptive behaviour based on criteria fixed at tutor creation time. Therefore, if later some different or more promising adaptation criteria are encountered it is compulsory to create a new tutor; so, the creation cost is duplicated. This generation methodology can be seen as static. It is an important drawback, as teachers usually show different teaching styles during a term, depending on the arising of new learning behaviours.

In this paper we present MAGADI, a learning platform that introduces a dynamic generation methodology, including the possibility of changing anytime the tutor adaptation criteria and the target student competencies. It is multi-domain, multi-competency and integrates different use phases. Moreover, it is flexible and can be easily extended. MAGADI is based on the IRIS authoring tool due to its domain independence -which makes it a good option to generate multi-domain systems- and our close knowledge about it.

1. MAGADI

A main goal of MAGADI is the generation of more *flexible learning systems*. This must be obtained from the use of a well integrated set of technologies and design structures that allow the future inclusion of new capabilities, such as administrative tasks or collaborative work. The *agent* concept is a powerful abstraction for the required characteristics[3], therefore, MAGADI has been developed as a multi-agent architecture that includes several *information*, *task* and *interface* agents as well as some *knowledge sources*. Its basic distribution into agents follows the component structure of the IRIS generated tutors', i.e. the INTZIRI[2] generic architecture which is supported by the classical three-component ITS architecture.

MAGADI[1] maintains the two IRIS use phases: authoring and learning. However, MAGADI is more flexible in the sense that both phases can occur anytime, providing a dynamic methodology for tutor generation. To allow this, the instructional planning knowledge is settled just at the beginning of the sessions, and the Student and Domain Models are only view-adapted to the teacher defined requirements. To make this possible, the instructional pedagogical requirements are stored in a Pedagogical Adaptation Database.

Figure1 shows the system architecture with some connected users. The *Receptionist Agent* waits for user connections and once the user has been identified, his or her set of agents is generated. For each user type the platform shows a different configuration, which shares the *Domain, Student and Pedagogical Adaptation* databases.

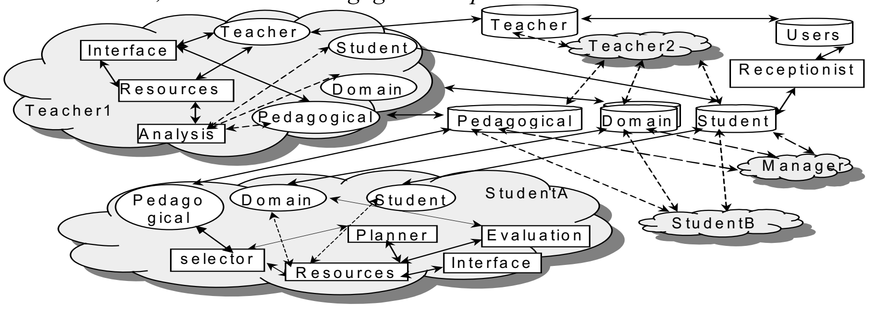


Figure 1. MAGADI Architecture

On each workspace, an *Interface Agent* allows the communication between the user and the system and a *Resources Agent* coordinates the agents of its dedicated workspace.

The teacher workspace allows defining the pedagogical requirements of the system, introducing domain information and following the student learning progress by visualising the information contained in the *Student Database*.

The student workspace allows students to follow different session types (free-exploring, guided or exercise-based sessions).

2. Authoring and Its Implications in Guided Student Sessions

MAGADI provides teachers with menu-driven graphical interfaces for authoring both the domain contents and the tutoring model. The domain contents can be introduced and modified incrementally, anytime, what provides a greater flexibility.

Authoring the tutoring model in MAGADI follows the IRIS approach, so the teacher defines the requirements the tutor will rely on to provide adaptivity. Thus, he can consider, or not, the student knowledge level, the type of didactic resource to be used, and so on; for example, he can decide that the system uses texts and no images for concepts. The requirements are stored in the *Pedagogical Adaptation Database* together with the target domain and student they will be applied to. Five assignation levels are provided, from the most specific: student-domain, group-domain, student, group and domain. This information is used by the student workspace agents to generate the *Planner* and the domain and student views. Student sessions begin with the student identification and the generation of default agents: *Resources, Pedagogical, Domain Student, Selector* and *Interface*. When the *Resources Agent* receives a message from the *Interface* agent asking for a guided session, it requests the *Selector* to generate the *Planner*; the last will include the appropriate set of instructional rules based on the requirements obtained from the *Pedagogical Adapter.* At the same time, the obtained requirement set is sent to the *Domain* and *Student* agents.

The *Pedagogical Adapter* searches the requirement set for the concrete student and domain in the *Pedagogical Database*, beginning from the most specific level. If none is found a generic one with default requirements is chosen.

Taking into account the selected adaptation style defined by the requirement set, the *Domain* and *Student* databases are masked to provide the desired knowledge views. This process is realised by specialised *Domain* and *Student* agents: they determine the set of attributes to be inspected and the information to be given for each information request. This way, those agents generate a partial view of the general models.

This view generation influences the system behaviour. On the one hand, the domain views produce the system to show only certain elements of the domain, facilitating the reuse of the domain among different system configurations. On the other hand, as the student view determines the student characteristics that will be used for pedagogical adaptation, the system adaptation style can vary from session to session as the selected requirements can vary among them –the teacher can define anytime new requirement sets.

3. Conclusions

This paper is centred on the issue of adaptivity in the context of distributed learning systems. Our approach to improve this aspect in a multi-session learning process is based on the concept of *dynamic generation of the tutoring knowledge according to the changing characteristic emerging for each session*. In this way, the same subject-matter can be learnt during several sessions with different learning styles. Moreover, the problems pointed by[5] to generate ILMS are solved on the basis of the inclusion of several domains and a general student model containing more information than that related to the pedagogical adaptation. This approach has resulted in the MAGADI platform.

A prototype of the platform has been developed and tested. Its validation with real students is foreseen during next term.

Acknowledgments: This work is supported by the Spanish Ministerio de Ciencia y Tecnología MCyT TIC2002-03141.

References

[1] A. Alvarez and I. Fernández-Castro, "An Open Adaptive and Multi-subject Educational System for the Web," *Frontiers in Artificial intelligence and Applications*, vol. 97, pp. 523-524, 2003.

[2] A. Arruarte, B. Ferrero, I. Fernández-Castro, M. Urretavizcaya, A. Álvarez, and J. Greer, "The IRIS Authoring Tool," in *Authoring Tools for Advanced Technology Learning Environments: Toward Cost-Effective Adaptive, Interactive and Intelligent Educational Software*, T. Murray, S. Blessing, and S. Ainsworth, Eds. Netherlands: Kluwer Academic Publishers, 2003, pp. 233-268.

[3] Jennings, "On Agent-Based Software Engineering," *Artificial Intelligence*, vol. 117, pp. 277-296, 2000.

[4] T. Murray, "An Overview of Intelligent Tutoring Systems Authoring Tools: Updated analysis of the state of the art," in *Authoring Tools for Advanced Technology Learning Environments: Toward Cost-Effective Adaptive, Interactive and Intelligent Educational Software*, T. Murray, S. Blessing, and S. Ainsworth, Eds. Netherlands: Kluwer Academic Publishers, 2003, pp. 491-544.

[5] K. Yacef, "Some thoughts about the synergetic effects of integrating ITS and LMS technologies together to the service of Education," presented at AIED 2003, Sydney, 2003.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Designing effective nonverbal communication for pedagogical agents

Amy L. BAYLOR
Director, Center for Research of Innovative Technologies for Learning (RITL) http://ritl.fsu.edu
Florida State University
baylor@coe.fsu.edu

Soyoung KIM
Instructional Systems Program
RITL – PALS
http://ritl.fsu.edu
Florida State University
syk02c@fsu.edu

Chanhee SON
Instructional Systems Program
RITL – PALS
http://ritl.fsu.edu
Florida State University
cs02k@fsu.edu

Miyoung LEE
Instructional Systems Program
RITL – PALS
http://ritl.fsu.edu
Florida State University
myl03@.fsu.edu

Abstract. This experimental study employed a 2x2x2 factorial design to investigate the effect of type of knowledge (procedural, attitudinal), deictic gesture (presence, absence), and facial expression (presence, absence) on learners attitudes, perception (agent persona, gesture, facial expression), and learning. A total of 237 participants learned from a instructional module that varied by the three factors. Results indicated that facial expressions were particularly valuable for attitudinal learning, and were actually detrimental for procedural learning outcome. Similarly, gestures were perceived as more valuable for students in the procedural module, even though they did not directly enhance recall.

1. Introduction

A particularly salient affordance of pedagogical agents is their propensity for effective message *delivery* in that manipulating its motivational [1] or affective content [2] can also dramatically impact learner beliefs and attributions. Thus, as a social interface, pedagogical agents can deliver messages through both verbal and nonverbal communications. While previous research has focused on agent verbal communication, for example, human voice vs. computer-generated voice [3-5], guidelines for agent nonverbal communication such as facial expression and gesture are nonexistent, particularly with respect for their relative value for *different* knowledge domains (e.g., procedural and attitudina). The purpose of this experimental study is to explore the effects of nonverbal communication (facial expression and deictic gesture) within procedural and attitudinal learning domains on attitude, learning, and agent perceptions.

2. Methods

Participants in this study included 237 undergraduate students (32.1% male and 67.9% female) enrolled in a computer literacy course in a southeastern public university. This study employed a 2×2×2 factorial design, with knowledge domain (procedural, attitudinal), deictic gestures (presence, absence), and facial expressions (presence, absence) as the three factors. The participants were randomly assigned to one of eight conditions and participated as a required course activity.

2.1 Independent Variables

Knowledge Domain – Learning Module. A procedural module was developed to teach participants how to perform specific procedural tasks associated with using a web-based software program intended to assess their proficiency in Microsoft Office applications. An attitudinal module was developed to elicit more desirable attitudes in students towards intellectual property rules and laws. The modules were designed to be equivalent with respect to rigor, time, and i mplementation of agent deictic gestures and facial expressions. Each agent (within a given module) differed only by its nonverbal communication, having identical scripts and machine-generated voice (to control for voice affect).

Deictic Gestures. Within the procedural module, deictic gestures were primarily used to indicate the physical objects and the geographical location of informative words on the interface. For the attitudinal module, deictic gestures directed the participants' attention to important information (e.g., to user interface features of the software). Approximately 50 instances of deictic gesture were incorporated in both types of instruction; there was no significant difference in number of gestures or distribution of them between the two modules.

Facial Expressions. The software Mimic2Pro was used to create facial expressions (neutral, serious, happy, surprised, and sad) and synchronize the expressions to the speech of the agents. Agents were designed to show appropriate emotion to link to the speech act. For example, when the agent talked about laws or rules, it displayed serious facial expressions whereas when it encouraged students to focus it exhibited happier expressions. Given the dynamic nature of the facial expressions, they could not be quantitatively compared across the modules, but were designed to be as similar in number and distribution as possible.

2.2 *Measures*.

The three sets of dependent variables included (1) attitude toward the content, (2) recall, and (3) agent persona. To assess learner attitude toward the content, learners were asked to list two adjectives that describe what they think about the copyright, scored as 3 for positive, 2 for neutral, 1 for negative, and 0 for no meaning. Recall was assessed by a 10-item test, consisting of true-of-false, multiple choice, and open-ended questions based on the content from the learning module. Recall questions differed for each module, but were developed in parallel format. Perceptions with respect to agent persona were assessed by the validated Agent Persona Instrument [6].

3. Selected Results

A three-way MANOVA was conducted to test the overall effects and a follow-up ANOVA was used for detecting each independent variable's effect. Table 1 below summarizes the results of the study.

Measures	Significant Results		
Attitude toward content	● Main effect of deictic gesture, $F(1, 229)=3.69$, $p<0.05$ (Deictic gesture: $M=4.53$, $SD=6.71$ vs. No deictic gesture: $M=4.11$, $SD=1.74$)		
	● Interaction between knowledge domain and facial expression, $F(1.229)=3.60$, $p<0.05$		
		Facial expression	*No facial expression*
	Procedural module	$M=4.12$, $SD=2.09$	$M=4.65$, $SD=1.69$
	Attitudinal module	$M=4.47$, $SD=1.56$	$M=4.19$, $SD=1.55$
Recall	● Main effect of knowledge domain, $F(1, 229)=245.45$, $p<0.05$ (Procedural module: $M=7.23$, $SD=1.68$ vs. Attitudinal module: $M=3.79$, $SD=1.60$)		
Agent Persona	● Main effect of facial expression, F (1,229) = 3.13, $p<0.1$ (Facial expression: $M=20.96$, $SD=6.14$ vs. no facial expression: $M=19.52$, $SD=6.25$)		
	● Main effect of knowledge domain, $F(1,229)=5.49$, $p<0.05$ (Procedural module : $M=21.06$, $SD=5.62$ vs. Attitudinal module: $M=19.39$, $SD=6.71$)		

4. Discussion

Results revealed an interaction effect between the knowledge domain of the learning module and the presence of agent facial expression, implying that students' attitudinal learning may be enhanced when agents have facial expressions. In contrast, student attitude toward the procedural content may be enhanced when agents have no facial expressions. The purpose of the facial expression for the procedural module was to encourage students to learn the verbal information which is inherently non-affective; thus the expressions were extraneous and may have unnecessarily overloaded the learners cognitively. In contrast, the agent facial expression within the attitudinal module was a more meaningful match, thus better situating the nonverbal communication to the information.

Results also indicated that participants rated the agent persona more positively when the agent had facial expressions (in either module). There was also a main effect indicating that the agent persona was rated more positively in the procedural module, perhaps because the agent's role was more as a conduit (e.g., directing student attention to interface features) rather than as a persuader in the attitudinal module. This also suggests that the domain of knowledge that agents portrayed impacts learners' perception of the presence of agents and the educational soundness of the agents' nonverbal communication. Consequently, instructional designers should consider the type of knowledge that they want to represent and transmit and then decide which type of nonverbal communication will effectively align with the type of knowledge. Overall, results from this study provide practical knowledge about the design of nonverbal communication for pedagogical agents to achieve positive outcomes, for both procedural and attitudinal learning. Unlike human nonverbal communication, agent animations can be designed and controlled to amplify the effect of the message and intensify its meaning in a more effective and efficient way.

5. Acknowledgments

This work was supported by the National Science Foundation, Grant IIS-0218692.

References

[1] A. L. Baylor, E. Shen, D. Warren, and S. Park, "Supporting learners with math anxiety: The impact of pedagogical agent emotional and motivational support," presented at Workshop on "Social and Emotional Intelligence in Learning Environments," held at the International Conference on Intelligent Tutoring Systems., Maceió, Brazil, 2004.

[2] A. L. Baylor, D. Warren, S. Park, E. Shen, and R. Perez, "The impact of frustration-mitigating messages delivered by an interface agent," presented at AI-ED, Amsterdam, 2005.

[3] B. Reeves and C. Nass, *The Media Equation*. Stanford, CA: CSLI Publications, 1996.

[4] R. K. Atkinson, R. E. Mayer, and M. M. Merrill, "Fostering social agency in multimedia learning: Examining the impact of an animated agent's voice," *Contemporary Educational Psychology*, vol. 30, pp. 117-139, 2005.

[5] A. L. Baylor, J. Ryu, and E. Shen, "The Effects of Pedagogical Agent Voice and Animation on Learning, Motivation and Perceived Persona," presented at ED-MEDIA, Hawaii, 2003.

[6] J. Ryu and A. L. Baylor, "The Psychometric Structure of Pedagogical Agent Persona," *Technology, Instruction, Cognition & Learning (TICL)*, in press.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Individualized feedback and simulation-based practice in the Tactical Language Training System: An experimental evaluation

Carole R. BEAL[1], W. Lewis JOHNSON[1], Richard DABROWSKI[2], & Shumin WU[1]
[1]*Information Sciences Institute, University of Southern California*
[2] *Instructional Systems Technology, Indiana University*

Abstract. Students worked with versions of the Tactical Language Training System for Arabic that varied in the provision of individualized feedback, and access to a simulation environment for practicing foreign language skills. Post test results indicated that feedback led to stronger learning outcomes. Students who worked with the complete TLTS rated it as comparable to working one-on-one with a human instructor.

1. Introduction

Computer simulation environments for learning have been associated with increased student engagement, yet to date there is only limited evidence that such experiences lead to stronger learning outcomes. One promising approach is to combine the individualized feedback provided in intelligent tutoring systems with the opportunities for practice and discovery provided in rich simulations [1]. This poster presentation reports the detailed empirical results of an experimental study evaluating the role of 1) individualized feedback and 2) a game-like simulation environment on adult students' learning of Levantine Arabic in an early version of the Tactical Language Training System (TLTS). The overall objective of the TLTS project is to promote rapid acquisition of task-related conversational skills in the less commonly taught languages [2]. Students practice their emerging foreign language skills by interacting with characters in a computer simulation environment, and receive individualized feedback on their pronunciation and performance.

2. Study methodology

The study included 21 soldiers at the Ft. Bragg military base located in the southern United States. Most (87%) had some prior experience with a language other than English, but none had learned Arabic. At the time of the study, the TLTS included three separate applications: the Skill Builder (SB), the Feedback module, and the Mission Practice Environment (MPE).

Skill Builder. The SB was an Arabic language tutorial designed around civilian affairs activities (e.g., repairing local infrastructure). The student learned to make proper greetings and introductions, use appropriate military ranks and titles, use directions (e.g., "right", "left", "go straight") and building names (e.g., "school", "hospital," "marketplace"), explain the mission (helping to make repairs to a local town infrastructure), to ask for the appropriate town official to contact, and to inquire where to find this person.

Feedback. The student practiced the Skill Builder exercises by speaking into a microphone attached to the computer. Individualized feedback on pronunciation was provided through audio recordings played into the student's headphones, based on a model of speech quality. For example, if the student's utterance for "MarHaba" ("hello") did not include the gutteral "H" sound, the student might hear "try dropping your tongue, "hhha""). Other feedback was more generic in nature and included motivational phrases such as "Good try" or "Try again" [3].

Mission Practice Environment. The MPE was a 3D simulation representing a rural Lebanese village. The student could speak in Arabic to animated village characters by recording into the microphone, and select gestures using the mouse wheel. The goal of the initial scene was to meet and question two men at the village coffeehouse as the source of the target information. If the student successfully established rapport via respectful greetings and introductions, and conducted a competent conversation in Arabic, the men would provide the name of and directions to the senior town official. The MPE thus allowed the student to practice the SB lesson content in a task-oriented conversational context.

Participants were randomly assigned to one of four versions of the TLTS. As shown in Table 1, all students worked with the SB, with some also using the MPE and/or receiving SB Feedback. Note that the "Hasan" students worked with the complete TLTS, including all three modules (SB, MPE, and Feedback). In contrast, the "Daud" students worked with a version that was most similar to off-the-shelf language tutorial software (SB only, no MPE, no Feedback).

"Hasan": SB, MPE, & Feedback	"Chalabi": SB & Feedback, no MPE
"Saiid": SB, MPE, No Feedback	"Daud" : SB, No Feedback, No MPE

Table 1: Experimental conditions: All students used the Skill Builder; groups varied by provision of individualized Feedback in SB, and opportunity to practice in MPE

Students worked with their version of the TLTS for four days, in one 90 minute session per day. On the fifth day, they completed a computer-presented post-test of Arabic proficiency and a survey about their impression of the TLTS. The post-test included vocabulary items, sentence comprehension and production (easy, difficult), listening comprehension (students heard an Arabic conversation and recorded an English translation), and speaking proficiency tasks (students participated in a simple, scripted Arabic conversation). All items were drawn from the Skill Builder lessons. More difficult items involved novel combinations of vocabulary and phrases.

3. Results and discussion

Arabic is a difficult language for most English speakers, and students' performance on the post test was not particularly good even after 6 hours of study. The highest post-test scores were observed for students who worked only with the Skill Builder and received personalized feedback on their speech. This was not particularly surprising because the post test was based on the Skill Builder content, and students who worked only with the SB had more time to devote to the lessons (i.e., they did not practice the material in the MPE simulation). Interestingly, however, the scores of the Saiid students matched those of the SB-only students on many post test measures. This suggests that the simulation might have led to more efficient learning, i.e., the Saiid students mastered the same SB lesson material in proportionally less time because they also spent time exploring the MPE. Also, it should also be noted that although the Saiid students did not receive individualized feedback in the

SB, they did receive some indirect feedback in the MPE when a village character did not understand their speech and would simply shrug in confusion.

Hasan students, who worked with the complete TLTS, scored relatively poorly on several post-test learning measures. However, these students were impeded by a higher rate of technical problems resulting from running three separate cycle-intensive applications (SB, MPE, Feedback) on fairly modest desktop computers. Even so, their comments were highly positive: "Although the program had bugs in it, it was still really good"; "I've learned a number of words just by playing the game"; "This new method of learning is very good"; "Gives us a break from the boring normal learning and goes more to a fun aspect of learning"; "In my experience, this is much more interesting and entertaining than sitting in a classroom". Hasan students also rated the TLTS as "about the same" as working one-on-one with a human instructor. In contrast, Daud students (those who worked with a version that resembled off-the-shelf language software) rated the experience as significantly less interesting and less helpful than other students, and rated their ablated version of the TLTS as less effective than whole class instruction.

Although the conclusions are somewhat limited due to the small group sample sizes, the results provide empirical support for the overall TLTS design and pedagogical approach, particularly the role of individualized feedback and the use of rich, interactive simulations for practicing and consolidating new skills. The Skill Builder, Mission Practice Environment, and feedback modules have subsequently been integrated into one seamless application. Future work focuses on the effectiveness of the complete TLTS for learners varying in language aptitude and motivational characteristics.

Acknowledgments

This project is part of the DARWARS Training Superiority Program of the Defense Advanced Research Projects Agency. The authors wish to acknowledge the contributions of the members of the Tactical Language Team. We would also like to thank Sgt. Jonathan Sirken of the 4th Psychological Operations Group at Ft. Bragg for his assistance with the study.

References

[1] Beal, C. R., Beck, J., et al. (2002, March). Intelligent user modeling and interactive entertainment. Paper presented at the American Association of Artificial Intelligence Spring Symposium, Stanford CA.

[2] Johnson, W. L., et al. (2004, August). Tactical language training system: An interim report. Proceedings of the 7th International Conference on Intelligent Tutoring Systems, Springer.

[3] Johnson, W. L., Wu, S., & Nouhi, Y. (2004, August). Socially intelligent pronunciation feedback for second language learning. Paper presented at the Workshop on Social and Emotional Intelligence in Learning Environments, 7th International Conference on Intelligent Tutoring Systems, Maceio Brazil.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Enhancing ITS instruction with integrated assessments of learner mood, motivation and gender

Carole R. BEAL, Erin SHAW, Yuan-Chun CHIU, Hyokyeong LEE, Hannes VILHJALMSSON, & Lei QU
Information Sciences Institute, USC Viterbi School of Engineering

Abstract. ITS instruction may be enhanced by models of student motivation and mood, in addition to cognitive skills and domain knowledge. In an initial study, self-assessments by high school students of their mathematics motivation and mood showed gender differences in response to ITS instruction, and predicted students' intention to learn from the ITS and use of multimedia help features.

Introduction

Much recent research points to the important role of student motivation in learning. Students who are highly motivated set goals, monitor their progress, evaluate their understanding, and use strategies to enhance learning, and have higher grades and test scores than less-motivated students. In fact, behaviors associated with high motivation are a stronger predictor of academic learning outcomes in some studies than measures of general intelligence [1]. Thus, adding a model of learner motivation should increase the pedagogical effectiveness of ITS instruction.

1. Project objectives

In this poster, we present our initial efforts to assess students' motivation and mood while working with an ITS. Self reports provided a reliable, non-intrusive and inexpensive source of motivation and mood data that could be easily collected in public school classrooms. Our initial target domain is high school mathematics, specifically, instruction in problem solving for high stakes achievement tests in the Wayang-West ITS. We were especially interested in the interaction of student gender and motivation. Much prior research indicates that females and males have different emotional reactions in mathematics, and that females have higher levels of test anxiety. Although females receive higher grades on average than males in math classes, females tend to score lower on high stakes achievement tests such as the SAT-M [2].

2. Study methodology

2.1 Participants. The study included students (N = 47) in two high school geometry classes in a large high school in urban Southern California serving a diverse student population. Students worked with the Wayang-West ITS during their mathematics class each day over one-week period, under the supervision of their classroom teacher.

2.2 Instruments. The Wayang-West ITS included integrated web pages at which students completed their daily "Math Personality Profile" (MPP). The MPP included instruments to assess a) beliefs about intelligence (fixed or possible to enhance), b) mood (anxious or relaxed/confident about the activity), c) mathematics motivation (self efficacy, liking of math, value of math), d) expected performance (predicted score on a real exam), e) intention to learn from the activity (attention, effort) and f) attribution for quality of the day's math work.

3. Results and discussion

Student motivation. Not surprisingly, students who had high self efficacy in math before the ITS intervention had higher expectations for success, felt that math was less difficult to learn, and predicted that they would get higher test scores than students with lower self efficacy (all correlations $p < .05$). The classroom mathematics teacher provided grade information and independent ratings of the students' observed mathematics motivation, which were highly correlated with students' self reports (correlations $p < .01$). Although most students were quite motivated and thought they were doing well, nearly half were performing below grade expectations. Thus, there were many students who wanted to do well and seemed to be trying, but who were not actually mastering the class material. This presents a pedagogical challenge: The ITS must be designed to raise students' objective skills while sustaining motivation (e.g., having high hopes and trying hard is not enough; acquiring specific strategies and skills is also critical).

Motivation and mood. Self efficacy was strongly correlated with the students' daily mood reports, and mood predicted students' specific estimates of their likely SAT-M scores, and the perceived difficulty of the task (i.e., more positive mood associated with lower perceived difficulty).

Gender comparisons. There were no gender differences in students' mathematics motivation or mood before the ITS activity started. However, by the end of the final session, males' mood had increased, whereas females' mood had significantly declined, even though there was no objective difference in ITS problem solving performance for males and females. Mood reports were also linked to male and female students' estimates of their test score on a real exam. Initially, students estimated that their score would be about at the national average, but males' increased their score estimates as they worked with the ITS, whereas female students' estimates declined. Thus, student gender is a potentially important factor to be considered in a pedagogical model: Male and female students were performing similarly, and both felt that the material was becoming easier to learn, yet their emotional responses diverged over time, as did their expectations for successful outcomes.

Use of multimedia help. Students' perception that math is difficult was a significant predictor of their use of help resources in the tutoring system, as was self-reported intention to learn. Not surprisingly, students' use of help resources (viewing multimedia hints) was negatively correlated with the number of incorrect answers entered (guessing or, "bottoming out"). Mood did not predict guessing behavior, but students who believed that people are born with a certain innate ability in math were more likely to enter multiple wrong answers per problem (i.e., to guess) than students with "incremental" beliefs about intelligence, $F(1,39) = 5.487$, $p < .05$. Providing praise for student performance can actually undermine math motivation, and increase beliefs that native talent is most important [3]. Thus, students may benefit most from ITS feedback focusing on their effort and use of the help resources as contributors to positive learning outcomes, rather than on feedback that emphasizes performance (number correct, scores relative to other students, etc.).

In a regression analysis with students' estimated test score as the outcome variable, mood and learning intention were both significant predictors, and mood accounted for a higher proportion of the variance. Thus, for example, students who started the final session feeling relaxed, confident and at ease expected to do better on the real exam, relative to other students who had similar intentions to work hard at learning but who felt more anxious, tense and worried. This suggests that daily mood assessments will be important to include in the enhanced pedagogical model of the ITS. For example, the student who is anxious might benefit from an initial review of problems that have already been tackled, along with feedback emphasizing incremental beliefs (e.g., "small steps add up").

A second regression analysis focused on students' estimates of their likely test performance after the ITS activity was over. Gender and mood at the start of the last session were both significant predictors, whereas factors such as the number of problems completed, and the use of help resources, did not account for significant variance. Again, it appears that students' affective state influenced their response to the ITS activity, with females showing less positive mood than males.

4. Conclusions and next steps

In this initial study, we established that students were able to report their motivational beliefs and affective states, using real-time self-reporting tools integrated into the ITS. Self reports were validated by ratings and grade information provided by their classroom mathematics teacher. The next step in the project is to implement our pedagogical model and select strategies appropriate for students who show high or low motivation, positive or negative affect, and so on. The pedagogical strategies that we are implementing are based on studies of how expert human instructors help students learn difficult material while sustaining motivation.

Acknowledgements

The Wayang Outpost ITS was originally developed with support from United States National Science Foundation HRD 0120809 [4]. The present project is supported by National Science Foundation grants HRD 0411532 and REC 0411886. The views represented in this paper do not necessarily reflect the position and policies of the funding agency. We would like to thank Ms. Devi Mattai, Ms. Jeanine Foote, Mr. Kenneth Banks, and Dr. Derick Evans of the Pasadena Unified School District for their support and assistance with this project. We would also like to thank Lewis Johnson and Mark Lepper for their assistance with the design of the motivational model.

References

[1] Zimmerman, B. J., & Schunk, D. H. (Eds.). (2001). Self regulated learning and academic achievement: Theoretical perspectives (2nd ed.). Mahwah NJ: Erlbaum.

[2] Beal, C. R. (1994). Boys and girls: The development of gender roles. New York: McGraw Hill.

[3] Mueller, C. M., & Dweck, C. S. (1998). Praise for intelligence can undermine children's motivation and performance. Journal of Personality and Social Psychology, 75, 33-52.

[4] Beal, C. R., Woolf, B. P., & Royer, J. M. (2001-2004). AnimalWorld: Enhancing high school women's mathematical competence. National Science Foundation HRD 0120809.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Exploring Simulations in Science through the Virtual Lab Research Study: From NASA Kennedy Space Center to High School Classrooms

Laura BLASI

Department of Educational Research, Technology, and Leadership (ERTL) The College of Education, The University of Central Florida (UCF) PO Box 161250 Orlando, FL 32816-1250, USA
lblasi@mail.ucf.edu

Abstract: Documenting the use of the Virtual Lab within nine high school biology classrooms in an urban school district in Florida, this study focuses on general science classes (i.e. not advanced or honors-level) within underserved populations over the 2004-2005 school year (n=225). The baseline data is presented from an administration of the Test of Science Related Attitudes (TOSRA) (Fraser, 1981) as the author shares preliminary analysis of the usability data in summer of 2005. With funding from the Bellsouth foundation, the researchers are contributing to the development of the 3d environment and the scanning electron microscope (SEM) simulation, in conjunction with the efforts of educational technology specialists at the Kennedy Space Center (KSC), NASA.

1. Introduction

Students make meaning and learn through the stories that contexualize the tools, skills, and facts that we give them. As Schank (1995) has observed, "Our machines do not solve puzzles, nor do they do mathematics. Rather, our aim is to make them interesting to talk to, an aspect of intelligence often ignored by computer professionals and intelligence assessors." Simulations provide access to the tools needed in high school classrooms for hands-on science exploration (c.f. Gordon and Pea, 1995) and this study aimed to document a baseline of data regarding the high school student population as well as the responses of students interacting with the simulation in a usability study contributing to the further development of the Virtual Lab. The Virtual Lab provides a navigable 3d lab environment run from the computer (http://education.ksc.nasa.gov/edtech/vl.htm). This project was funded through the NASA Learning Technologies Project and targets high school and entry-level college students. The software provides an environment with enough information and realism to give students the experience of operating the actual SEM instrument. Prior studies have explored the impact of the use of simulations in science on students achievement and attitudes (Huppert, Lomask, and Lazarowitz, 2002; Geban, Askar, and Ozkan,1992) with positive results, as the simulations allow students repeated practice with access to sophisticated equipment.

2. The Methods

This study focused on tenth graders, students approximately 16 years old, at a time when neuroscience has shown an increased capacity for scientific reasoning, at the stage of development students enter after their middle school years (Kwon and Lawson, 2000). The teachers who participated, leading the nine high school general science (not advanced) biology classrooms, were teaching in the schools for a minimum of one year prior to the study. At each of the three schools there was one control group (C) and two treatment groups: one with the technology and training (B) and another with technology, training, and assistance (A). The sample (N=225) was randomized at the classroom level. An experimental design was used to document the impact – if any – of the researcher interactions within the classrooms (A), measuring change by a pre- and post-test administration of the Test of Science Related Attitudes (TOSRA) (Fraser, 1981). Standardized achievement test items in science from the fifth grade and the tenth grade tests were used to document student performance in biology across the three schools. Grounded in this baseline of data regarding attitudes toward science, test item performance, and demographic data, the usability study focused on Group A. Researchers spent time in classrooms documenting students' on screen use of the technology (video) as well as their voices (audio), as they narrated their reactions and experience navigating the software program without interruption (Dumas & Redish,1993). Cognitive interviewing techniques were used to document student perspectives in response to specific questions during their software use (Ericsson & Simon, 1993).

3. Findings and Further Research

While the usability data is being analyzed in the summer of 2005, the preliminary findings provide some insight into the conditions for developing and implementing simulations in general science classrooms at the high school-level in the US. This information is especially important for developers who may not have classroom teaching experience and may not be able to interact with students in the content areas. The demographics reveal that of the three schools, school 2 had the highest population of students who were not white (78%), and the highest number of students who were eligible for federal assistance through the "free or reduced lunch" program (53%).

The interviews with the teachers documented their interest in using tools in the science curriculum alongside observation in order to document student achievement. As one explained: "The best way I can measure some type of gain in learning it's usually with some type of hands-on activity, which requires them to do some type of performance tasks in front of me…you can cover microscope use...preparing the slides…knowing how to manipulate the microscope to see a better picture...." (Sarah, 10th grade biology teacher). However a consistent theme across interviews was confirmed by the research team observations in all three schools: science classes in this district have little or no access to computer labs due to the emphasis on preparation for standardized tests in other subjects, while at the same time classrooms are rarely equipped with computers beyond the teacher's desk and the hardware donated for participation in this research study.

A small sample of achievement test items in biology were administered from fifth grade and tenth grade exams at the beginning of the study. Analyzing only the performance on fifth grade items in fall 2004, these tenth grade students in schools 1 and 3 scored correctly on 60% of the items, while students in school 2 scored correctly on 48% of the items. This performance matches the ratings by the state based on overall standardized test results, which rank school 2 the lowest of the three. Using the TOSRA (Fraser, 1981), a 70-item self-report instrument with seven dimensions, researchers documented: a) attitudes towards careers in science; b) evidence of scientific attitudes; c) evidence of application of scientific inquiry as measured by a pre-post administration of the TOSRA. A higher mean score (mean scores can range between 1 and 5) is indicative of a more positive view of science and the pretest documented that student scores were not significantly different across schools, despite differences in demographics and performance on state standardized achievement tests (mean=3.1 out of a possible 5). Overall, observations and interviews across all three schools revealed that the conditions for implementing classroom simulations are not optimal. The schools varied in terms of demographics and achievement performance, however the student scores on the TOSRA were not significantly different, with an average score of 3 out of 5 on the 70-item instrument.

While conducting an analysis of the usability data, the author will work at NASA KSC in the summer of 2005. The long term goal of the study is to contribute to the further development of educational software and simulations at NASA KSC, with high school students who are in general science tracks as the primary audience. This report was written as a Fellow in the Academy for Teaching and Learning at UCF.

References

Dumas, J.S., & Redish, J.C. (1993). *A practical guide to usability testing*. Norwood, NJ, Ablex.

Ericsson, K. A. & Simon, H. A. (1993). *Protocol analysis: Verbal reports as data.* Revised edition. Cambridge, MA: The MIT Press.

Fraser, B. J. (1981). *TOSRA. Test of Science-Related Attitudes handbook.* The Australian Council for Educational Research Limited.

Geban, O., Askar, P., & Ozkan, I. (1992). Effects of computer simulations and problem-solving approaches on high school students. *Journal of Educational Research, 86*(1), 5.

Gordon, D.N. and Pea, R.D. (1995). Prospects for scientific visualization as an educational technology. *Journal of Learning Science 4*(3), 249–279.

Huppert, J., Lomask, S. M., & Lazarowitz, R. (2002). Computer simulations in the high school: Students' cognitive stages, science process skills and academic achievement in microbiology. *International Journal of Science Education, 24*(8), 803-821.

Kwon, Y. J., & Lawson, A. E. (2000). Linking brain growth with the development of scientific reasoning ability and conceptual change during adolescence. *Journal of Research in Science Teaching, 37*(1), 44-62.

Schank, R. C. (1995). *Tell me a story: Narrative and intelligence*. Evanston, Ill.: Northwestern University Press.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Generating Structured Explanations of System Behaviour Using Qualitative Simulations

Anders BOUWER and Bert BREDEWEG
University of Amsterdam, Human-Computer Studies Lab
The Netherlands
Email: {bouwer,bredeweg}@science.uva.nl

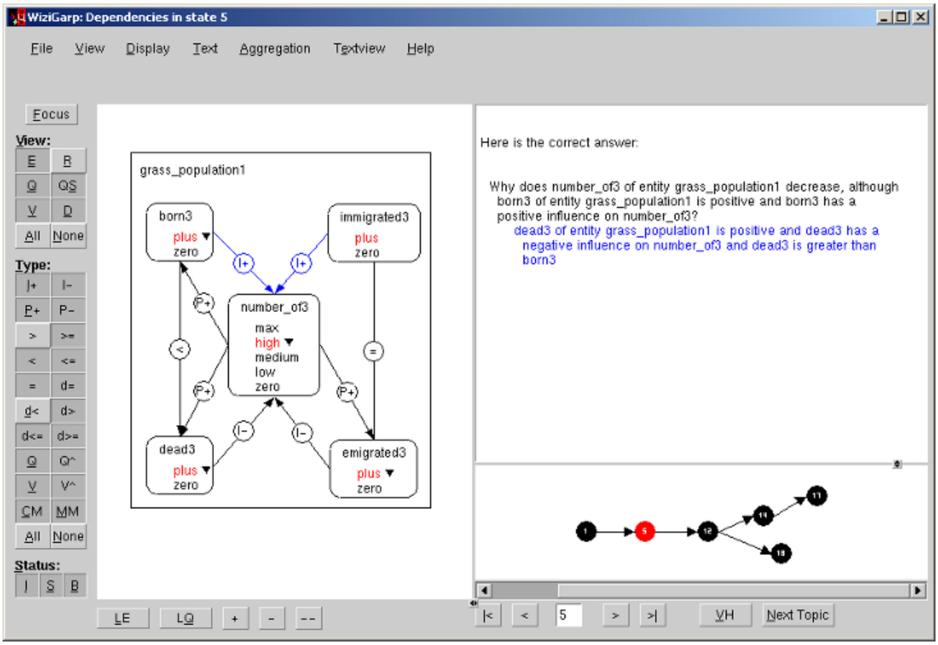
Abstract. This paper presents an approach to generate structured explanations of system behaviour based on qualitative simulations. This has been implemented in WiziGarp, a domain-independent learning environment. The main research question addressed here is how to manage the complexity of the simulations in order to generate adequate explanations.

1. Introduction

Qualitative simulations explicitly represent the kinds of knowledge that can support learners in building their own conceptual model of dynamic phenomena. This knowledge is used to generate a state-transition graph of all possible behaviours. The main problem in using qualitative simulations in education is that due to the amount of detail included, a simulation can be very complex, *i.e.*, containing a large number of states and transitions, and a large amount of information within each state.

2. Aggregation of Qualitative Simulations

In order to organize the information to be communicated, five *levels of aggregation* are introduced, which vary from the individual system state level via longer time-frames to the global level, at which alternative possibilities occur. For each of these levels, aggregation techniques have been implemented which reduce the amount of information to be communicated by the WiziGarp system. On the system state level, the status of causal dependencies is analyzed to arrive at a *classification of causal effects*, as inactive, submissive, balanced, or effective. This allows grouping and selection of those dependencies which have an actual effect, discarding dependencies whose effect does not contribute to the outcome of the simulation. The classification refines work by De Koning et al. [2], who distinguish only between submissive and effective dependencies, and work by Mallory [5], who attributes similar labels directly to quantities instead of to their effects. Figure 1 shows an example screenshot from WiziGarp with a subset of dependencies for a grass population in Brazilian Cerrado ecology [6], including submissive dependen-



Here is the correct answer:

Why does number_of3 of entity grass_population1 decrease, although born3 of entity grass_population1 is positive and born3 has a positive influence on number_of3?

dead3 of entity grass_population1 is positive and dead3 has a negative influence on number_of3 and dead3 is greater than born3

Figure 1. A WiziGarp screenshot: dependencies for the grass population and an exercise question

cies (the positive influences), as well as effective ones (the negative influences and the positive proportionalities). On the local event level, *recognition of events* is performed. Information from adjacent states is selected and combined to form larger chunks of information, specifying meaningful events of various types, such as value and derivative events (e.g., Q_x starts to increase), causal effect events (e.g., the influence from Q_x on Q_y becomes inactive), and model fragment events (e.g., the model fragment for a particular process becomes active). On the path (segment) level, additional value and derivative events are recognized by selection and chunking of lower level events (e.g., the highest value of Q_x that is reached in the path (segment) P is V, or Q_x fluctuates between V_1 and V_2, respectively). On the global level, *transitive reduction* and *aggregation of alternative orderings* are used to simplify the state-transition graph. Transitive reduction abstracts from all transitions T ($= S_x \rightarrow S_y$) for which holds that there is a path P from S_x to S_y which does not contain transition T, with the extra condition that P contains the same events as T. Aggregation of alternative orderings abstracts paths which divert and reunite, if they include the same events, albeit in a different order. The algorithms for these techniques can be found in [1]. In the figure, the state-transition graph shown is the result of performing aggregation of alternative orderings. It contains 6 states and only 2 paths, whereas the original state-transition graph contained 19 states and 869 possible paths. This reduction is possible because most paths contain the same events and only differ in the order in which the events occur.

3. Generating Interactive Explanations

WiziGarp can present topics using various didactic means: diagrams, textual descriptions, causal explanations, contrastive explanations, queries or exercise questions. Which ones are used in what order, as well as the desired level of aggregation for each topic, can be specified in a didactic plan (these are currently handcrafted). WiziGarp can take the initiative by asking exercise questions that are automatically generated, as described in detail in [4]. Figure 1 includes a question generated about the grass population. The learner can also take the initiative and ask for specific information to be answered by WiziGarp. To this end, events on the local level and path (segment) level are determined for the quantities of interest. The learner can ask queries about a particular event by selecting it and choosing a query from a popup menu that arises. For example, when the learner asks why the number of trees has started to increase, this is causally explained by the introduction of the effect of immigration. In addition, contrastive explanations can be generated which highlight the differences between states, or paths.

4. Discussion

Several parts of the WiziGarp architecture have been evaluated by potential users and domain experts, such as the question generation module [4], and the diagrammatic representations [1]. The results of these evaluation studies are encouraging and have informed the design of the other modules. The approach is generic and has also been tested on other domains, such as physics and biology. Compared to related work that addresses explanations for simulations [3, 5], WiziGarp encompasses a more extensive taxonomy of events, more flexible aggregation mechanisms, and a richer set of didactic means. Future work will address reactive curriculum planning, based on learner answers.

References

[1] A. Bouwer. *Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments*. PhD thesis, University of Amsterdam, to appear.

[2] K. de Koning, B. Bredeweg, J. Breuker, and B. Wielinga. Model-based reasoning about learner behaviour. *Artificial Intelligence*, (117):173–229, 2000.

[3] K. D. Forbus, P. B. Whalley, J. O. Everett, L. Ureel, M. Brokowski, J. Baher, and S. E. Kuehne. Cyclepad: An articulate virtual laboratory for engineering thermodynamics. *Artificial Intelligence*, 114(1/2):297–347, 1999.

[4] F. Goddijn, A. Bouwer, and B. Bredeweg. Automatically generating tutoring questions for qualitative simulations. In P. Salles and B. Bredeweg, editors, *Proceedings of QR'03, the 17th Int. workshop on Qualitative Reasoning*, pages 87–94, Brasilia, Brazil, 20-22 Aug. 2003.

[5] R. S. Mallory. *Tools for Explaining Complex Qualitative Simulations*. PhD thesis, Department of Computer Sciences, University of Texas at Austin, 1998.

[6] P. Salles and B. Bredeweg. Constructing progressive learning routes through qualitative simulation models in ecology. In G. Biswas, editor, *Proceedings of QR'01, the 15th Int. workshop on Qualitative Reasoning*, pages 82–89, San Antonio, Texas, 17-19 May 2001.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

The Bricoles project: support socially informed design of learning environment

Pierre-André Caron, Alain Derycke, Xavier Le Pallec
Laboratoire Trigone, équipe NOCE
Université des Sciences et Technologies de Lille
59655 Villeneuve d'Ascq cedex - France
pa.caron@ed.univ-lille1.fr

Abstract: In this paper we describe our current work on the BRICOLES project. Its objective is to provide an environment which helps teachers to elaborate e-learning courses. We show how MDD approach could be apply to e-learning creation. We use RAM3, a meta-modeling tool which is developed in our laboratory. With RAM3, we express pedagogical scenario and study models define on different e-learning platform metamodel. Then we export model on targeted platform with scripts.

Introduction

The BRICOLES [**B**ring off **R**eflexive, **I**ntuitive and **C**onceptual **O**pen **LE**arning **S**ystem] project main objective is to suggest solutions to reintroduce teacher in e-learning courses design.

The design of the Bricoles project with a Model Driven Approach

Using artifacts[1] and "Bricolage"[2] are two natural ways to help teacher to design distant courses. We chose to adopt ideas from model driven engineering to "materialize" such concept/method. Broadly speaking, using model driven tools begins by defining a logical model, without technical/implementation details, and ends by automatic generation of corresponding application (after selecting the implementation platform). Graphical Models, which are manipulated provide better boundary objects and models transformation allows "Bricolage" by reusing experience of others and by adapting models to target platform[3]. Supporting different modeling formalisms is complex to implement and models transformation generally needs to define transformation rules which are not easy. In our context, fortunately these rules are not defined by teachers. We use Model Driven Approach, by defining the corresponding metamodel for each e-learning platform and implement deployment facility which will be fed by instances of previous metamodels. Then, we put IMS-LD [4] forward for modeling. This pedagogical metamodel represents a standardization effort from the educational community), it allows teachers to express easily dependencies between pedagogical intention and platform functionality.

Life cycle

More than a tool, our proposition consists in a design environment which is composed by two tools: RAM3 [5] - to support different modeling formalisms; GenDep - to deploy model on e-learning platform.

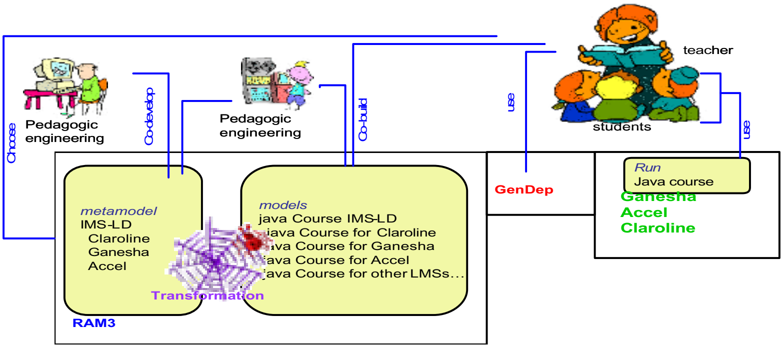


Fig 1 life cycle

Figure 1 represents roles of each hypothetical user of our environment. Computer scientists and pedagogical engineers define, with RAM3, modeling formalisms like IMS-LD or Ganesha's metamodel. Teachers or pedagogical engineers define, with RAM3, models of pedagogical scenarios according to metamodels defined before. With GenDep the teacher deploys an instance of a course (play the scenario) in a e-learning platform (Ganesha [6]).

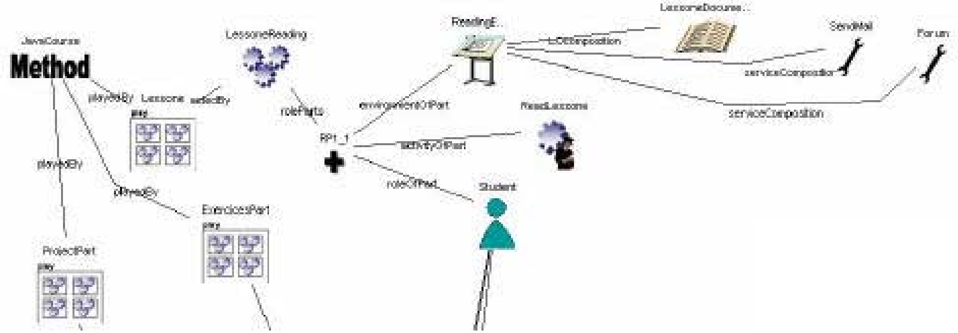


Fig 2 Course java

We illustrate the use of these tools for a teacher who wants to teach a Java course. The Java teacher defines the scenario. He begins to load IMS-LD in RAM3. Then, he may define the IMS-LD model corresponding to the Java course (Fig 2). He describes the different roles (students, author, and teacher) and the different phases (to study documents, to do exercises, to realize small project) which may run in parallel. Next, he chooses to project his model to the Ganesha platform (he can do this because transformation rules has been defined before). The transformation engine creates the Ganesha model corresponding to the previous IMS-LD model. The teacher may use RAM3 to edit the resulting Ganesha model in order to improve or refine it. This transformation/exportation reveals differences between the IMS-LD scenario and the Ganesha one. Finally, teacher uses GenDep to deploy his Ganesha model on the platform (Ganesha) where it has to do his course. GenDep asks him web address of the platform, it simulates a web user filling web forms, which are presented by the platform, in order to deploy corresponding group, to assign students to group, to allocate resources to students…. Simulation is done by sending HTTP requests (protocol used on Internet). We are studying same process to export our scenario on Claroline, Moodle and PostNuke platforms [7].

Conclusion

Modeling is the main principle of our proposition. It softens the transition between needed pedagogical bricolage and needed computer structural data. For different projects (like European Kaleidoscope [8]) we have written several scenarios and modeled them. We now want to specify metamodel of other platforms and their associated “deployment protocol”. We are developing a graphical editor to define transformation rules and we are studying other artifacts.

Reference

[1] Wartofsky, Marx W (1973), Perception, representation, and the forms of action: toward an historical epistemology, in Wartofsky, M. W., Models, Dordrecht: D. Reidel Publishing Company, 1979.

[2] Levi-Strauss, C., (1966). The Savage Mind. (2nd. ed.) Chicago: University of Chicago Press.p 17

[3] Büscher Monika , Gill Satinder, Mogensen Preben and Shapiro Dan, Bricolage as a Method for Situated Design, Landscapes of Practice

[4] Koper, R. Modeling Units of Study from a Pedagogical Perspective – The Pedagogical Meta-Model behind EML. Open University of the Netherlands. First Draft, Version 2, 2001.

[5] Le Pallec Xavier, Derycke Alain, RAM3: towards a more intuitive MOF meta-modelling process, SCI 2003, Systemics, Cybernetics and Informatics.

[6] http://www.anemalab.org/ganesha/

[7] http://www.claroline.net/ , http://moodle.org/ , http://www.postnuke.com

[8] http://www.noe-kaleidoscope.org/

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Explainable Artificial Intelligence for Training and Tutoring

H. Chad LANE, Mark G. CORE, Michael VAN LENT, Steve SOLOMON, Dave GOMBOC
University of Southern California / Institute for Creative Technologies
13274 Fiji Way, Marina del Rey, CA 90292 USA
{lane, core, vanlent, solomon, gomboc}@ict.usc.edu

Abstract This paper describes an Explainable Artificial Intelligence (XAI) tool that allows entities to answer questions about their activities within a tactical simulation. We show how XAI can be used to provide more meaningful after-action reviews and discuss ongoing work to integrate an intelligent tutor into the XAI framework.

Introduction

Military training aids typically provide an after-action review (AAR) tool to allow students to review their exercises and ideally learn from them. Common features of these tools include mission statistics, a list of accomplished and failed objectives, and sometimes a mission replay feature. Because of increasingly complex artificial intelligence (AI) in these training aids, it has been difficult for users of such AAR tools to understand how their orders translate into the activities of computer-controlled entities with such limited AAR tools. Student users have the additional disadvantage of possessing fragmented and possibly flawed domain knowledge: they are faced not only with learning new tactical knowledge (i.e., how units perform their tasks) and new skills (i.e., constructing plans and updating them on the fly), but also comprehending emergent behaviors and their triggers.

To provide a better AAR tool and to help users better understand entities' actions in military simulations, we have developed a portable Explainable AI (XAI) module that allows a user to question entities directly about their actions, status, and goals. For live training exercises, the U.S. Army defines AARs as "a review of training that allows soldiers, leaders, and units to discover for themselves what happened during the training and why. It is also used to solicit ideas on how the training could have been performed better" [2, App. G]. During such AARs, participating soldiers are usually available for questioning. The goal of our research is to use XAI to provide this feature for computer-controlled soldiers.

1. Explainable Artificial Intelligence for Training

Although our current XAI system was not designed exclusively for pedagogical purposes, it nonetheless addresses the first part of the AAR definition given above: it allows for a deeper investigation of what happened during an exercise. To interact with entities, the student may select from time points of focus, a particular entity to question, and finally a question to ask. Some questions are specific to a particular entity (e.g., what is your health?) while others concern a larger group (e.g., what is your unit's task?). Time points and available questions are dependent on the entity in focus and the discourse context and exchanges with the system are displayed as question and answer dialogues. More details about the interface and system implementation can be found in [3].

A sample interaction with our XAI system appears in Figure 1. The student begins by asking questions of Gligoric, an opposing force entity (OPFOR). Later, the student changes to Evans, a blue force entity (BLUFOR). This dialogue is intended to show how a student might investigate an event using the XAI tool. The student here probably believes Evans' unit was somehow involved in the death of Gligoric and is examining the events leading up to it. Although some of this information could potentially be derived from a graphical replay (we have provided this capability in previous work), there is a chance it will not be obvious to a student from the replay (e.g., line 9). Such questioning can also therefore be used by the student to confirm inferences made from visual evidence alone. In its current state, XAI is best viewed as a *discovery environment*. That is, the student is free to use the tool as desired to explore the simulation results and achieve the goals of the AAR. Having just completed the exercise for "real," interactions, such as the one in the figure, give the student a chance to learn more about the entities and what they experienced. It is up to the student, however, to ask the right questions of the right entities and understand the responses.

1. User: What is your health/damage status?
2. Gligoric [2:26]: **I am unhurt.**
3. User: Let's fast forward to simulation time 4:16.
4. Gligoric [4:16]: **I am currently withdrawing.**
5. User: What is your health/damage status?
6. Gligoric [4:16] **I have been killed in action.**
7. User: Let's talk to Private Evans.
8. Evans [4:16]: **I am the fire team leader of Blue Fireteam 2. My unit's task is to clear the W room of the post office.**
9. User: Let's rewind to simulation time 2:32
10. Evans [2:32]: **I am currently transiting to the right stack position outside of the NW door of the W room of the post office.**
11. User: What are you shooting at?
12. Evans [2:32]: **I am shooting at an entity located at 125 m NNE of the NW door door of the W room of the post office.**

Figure 1. Sample XAI dialogue

Focusing more specifically on our system's dialogue manager and natural language generator, we see that pedagogical support is built into these components. Currently we maintain a simple dialogue state consisting of all the entities and units that the user has talked with. In the dialogue in figure 1, Evans introduces himself as fire team leader and describes his unit's task because the student has not talked with either Evans or anyone else in that unit. This feature is a placeholder for more powerful reasoning about how to adapt the system's output to the student (e.g., it should not use undefined technical terms, it may need to explicitly state knowledge implied by its explanations). Although it is currently simulation-dependent, our system also maintains specific points of reference to refer to when responding to questions that require some location-oriented answer (e.g., line 12 in the Figure 1).

2. Related Work

The motivation for and technical challenges of explaining the internal processing of AI systems have been explored at length in the context of medical diagnosis systems. One prominent example, MYCIN, used a complex set of rules to diagnose illness and suggest treatments based on patient statistics and test results [6]. The developers of these systems were quick to realize that doctors were not going to accept the expert system's diagnoses on faith. Consequently, these systems were augmented with the ability to provide explanations to justify their diagnoses. Education becomes a natural extension as well since explanation is often an important component of remedial interventions with students. Three notable efforts falling into this category are the Program Enhancement Advisor (PEA) for teaching LISP programmers to improve their code [5], the family of successors to MYCIN [1], and another entity-driven explanation system, Debrief [4].

3. XAI for Tutoring

Evidence for learning in pure discovery environments is marginal [5], and so we are in the early stages of designing an intelligent tutoring module with the goal of providing a more guided discovery experience for students. We adopt the general goals of an AAR: review what happened, investigate how and why these events occurred, and discuss how to improve future performance. Answering *why* questions is a significant technological challenge, but also highly relevant to good tutoring. For example, discovering why a unit has paused in the middle of executing a task has the potential to help a student who gave the order to proceed. This may require reasoning about previous or concurrent events in the simulation. If a unit is currently under fire, for example, it is critical that the student understand what has caused the delay. It could very well involve an earlier mistake, such as failing to provide cover. The student could be asked to analyze the situation and suggest ways to allow the unit in question to proceed. One such question would be "Now that you have learned why this unit is delayed, what was missing from your plan?" If the student cannot generate any ideas, hints such as "Can you think of a way to conceal the unit for safe movement?" or "Do you see any other nearby units that could provide cover fire?" would be appropriate. We hypothesize that questions such as these, and more dynamic AARs, will improve students' self-evaluation skills and problem solving abilities within the simulation.

In addition to working with tactical behaviors, we are also in the early phases of targeting non-physical behaviors, such as emotional response, for explanation. This has advantages for systems that aim to teach subjects such as negotiation skills, cultural awareness or sensitivity. Explaining why an utterance (by a user) has offended an automated entity is, for example, similar to explaining emergent tactical behaviors. Tutoring in situations like this would, we believe, also be similar (e.g., "Could you have phrased that differently?").

Acknowledgements

The project or effort described here has been sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM). Statements and opinions expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

References

[1] Clancey, W. J. (1986) From GUIDON to NEOMYCIN and HERACLES in twenty short lessons, *AI Magazine*, Volume 7, Number 3, pages 40-60.

[2] FM 25-101. (1990) Battle Focused Training. Headquarters, US Dept. of the Army. Washington D.C.

[3] Gomboc, D., Solomon, S., Core, M. G., Lane, H. C., van Lent, M. (2005) Design Recommendations to Support Automated Explanation and Tutoring. To appear in *Proceedings of the 2005 Conference on Behavior Representation in Modeling and Simulation (BRIMS)*, Universal City, CA. May 2005.

[4] Johnson, W. L. (1994) Agents that learn to explain themselves. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 1257-1263.

[5] Mayer, R. M. (2004) Should There Be a Three-Strikes Rule Against Pure Discovery Learning? *American Psychologist*, Volume 59, Number 1, pages 14-19.

[6] Shortliffe, E. H. (1976) Computer-based Medical Consultations: MYCIN. Elsevier, New York.

[7] Swartout, W. R., Paris, C. L., and Moore, J. D. (1994) Design For Explainable Expert Systems. *IEEE Expert*, Volume 6, Number 3, pages 58-64.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

An Agent-based Framework for Enhancing Helping Behaviors in Human Teamwork

Cong CHEN, John YEN
The School of Information Sciences and Technology, the Pennsylvania State University
University Park, PA 16802, U.S.A.
Michael MILLER and Richard VOLZ
Department of Computer Science, Texas A&M University
College Station, TX 77843, U.S.A.
Wayne SHEBILSKE
The Department of Psychology, Wright State University
Dayton, OH 45435, U.S.A

Abstract. This paper proposes an intelligent training framework where agents are used with explicit teamwork models for desired teamwork behaviors. In the framework, we divide coaching process into two manageable sub-phases and model trainees regarding teamwork dynamics and team performance. We have implemented the framework on a team-based agent architecture (CAST) and applied it to train helping behaviors for a simulated command and control (C2) task. The framework and its implementation enable us to design experiments for studying the effectiveness of agent-based coaching for helping behaviors among team members.

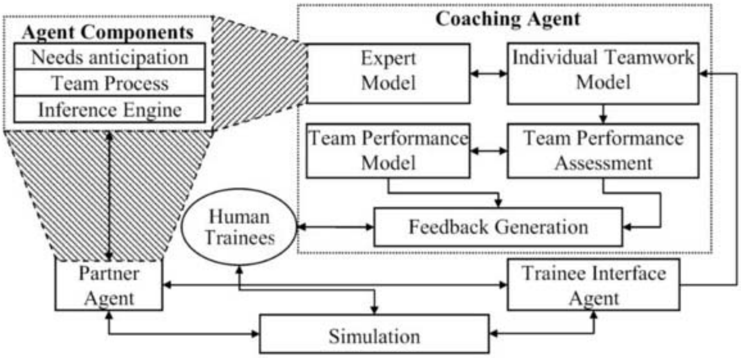
Introduction

The objective of this research is to build an agent-based team training system to coach human trainees, specifically facilitate trainees' learning of how to help each other when collaboratively achieve a mission. In section 1, we propose an agent-based intelligent team training (AITT) framework that supports coaching of helping behaviors. In section 2, we describe the design of the coaching agents to be used in the framework, focusing on the two-phase training protocol. Discussions and conclusions are given in section 3.

1. An Agent-based Intelligent Training Framework

CAST-ITT (CAST Intelligent Team Training) system is a team training system extended from a multi-agent infrastructure CAST [1]. A generic Agent-based Intelligent Team Training (AITT) framework has been developed to monitor the interactions among intelligent agents, human trainees, and the simulation system. Agents serve multiple roles in AITT – one role is to be the virtual partners who perform similar tasks - components of a *partner agent* are shown on the left side of Figure 1; the other role is to support user modeling of teamwork and provide coaching feedback to the team regarding team members' helping behaviors – components of a *coaching agent* are shown on the right side of Figure 1. To build the coaching agents in the framework, we utilize user modeling components and reinforce a set of human training strategies. Both *overlay model* and *error taxonomy* are used to diagnose trainee errors.

Figure 1 Agent-based Intelligent Team Training Framework



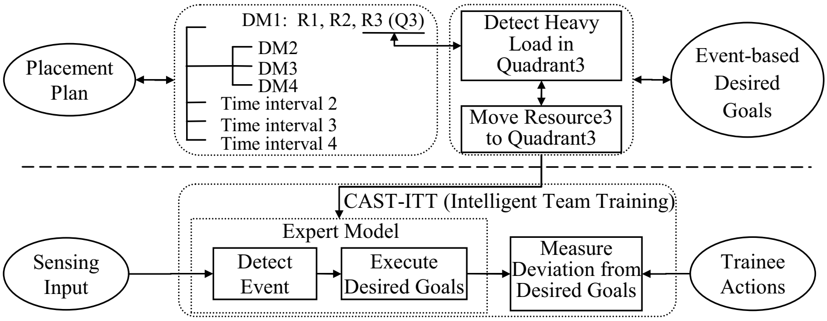
The *individual teamwork module* captures information relevant to teamwork related behaviors for individual human trainees. The e*xpert model* represents both the expert strategies acquired from the domain experts or the qualified trainee knowledge after planning phase evaluation. The *team performance assessment module* generates team performance measures and the assessment results. A *team performance model* of trainees is maintained to facilitate the reasoning of team performance history, which distinguishes our work from the traditional assessment module in a user modeling framework.

2. Two-phase Coaching

We choose the Distributed Dynamic Decision-making (DDD) simulation as the domain to be used with the CAST-ITT system. The team mission of DDD is for a team to collectively protect restricted zones [2]. The experiment is set up so that particular team members are overloaded and the mission can only be achieved when they assist each other. We coach team collaboration via two sub-phases—in the *planning phase,* agents coach on trainee's planning of team collaboration and in the *execution phase,* agents coach on team's collaboration process.

In the planning phase, we provide trainees additional information through the use of an *intelligence report* and allow them to plan the allocation of team resources. *Intelligence report* gives an overview of the track arrival information. Via our graphic planning tool, trainees communicate, make decisions and come up with a *team placement plan* about how to allocate their vehicles. A scoring algorithm has been developed to evaluate trainees' placement plan based on a list of prioritized expert strategies, with respect to trainees' resources and work loads. A higher score indicates a better helping pattern in the planning phase. Feedbacks are categorized into the major domain tasks, such as identification or attack of tracks. Possible errors can be trainee's planning too much help without covering own zone, or planning no help when extra vehicles can be sent to particular teammates. In the execution phase, trainees' online performance can be measured based on the dynamic sensing information, trainees' collaborative actions and the qualified placement plan generated in the coaching of planning phase. When a helping event is triggered, there isn't a precise "desired behavior" for the team; how the team performs depends on many domain factors and good strategies have to be adaptive to execution contingencies. Without having a comprehensive expert model, we use a hybrid approach of *overlay model* and *error* *taxonomy* – modeling only trainee's high level goals and diagnosing only the trainee errors related to several critical factors of a helping event. Figure 2 shows the generation of the event-based target goals in the *planning* phase, the execution of these desired goals in coaching agent's expert model and the evaluation of trainee's performance by comparing trainee actions with the target behaviors in the *execution* phase. Each trainee in the domain acts an individual Decision Maker (DM).

Figure 2 Two-Phase coaching: Planning and Execution of Team Collaboration



3. Discussions and Conclusions

In this paper, we report our ongoing research of using intelligent agents to coach helping behaviors among team members. To avoid the costly monitoring of trainee behaviors throughout a task, we propose an efficient user modeling system that only captures the high-level desired goals for the target knowledge and skills. Feedback is provided to the trainee at the end of each session when an overall comprehensive assessment can be made. The information we collect from the trainee might be domain dependent, yet the domain independent nature of helping behaviors enables the AITT framework to be used by other researchers to design and test coaching applications within a teamwork-based domain – the framework can fit in other agent architectures or apply to multiple domains with the inclusion of our coaching agents.

It is not always an advantage when humans have to make decisions within an automated interactive system [3]. Agents can be very concise and accurate about a specific set of rules while humans are good at adaptation to domain contingencies. We have observed that trainees and agents can take advantages of their unique strengths by acting in a complementary manner. To allow positive interactions outside the automated system, human trainees are explicitly encouraged discussing about their own helping strategies with the necessary information and feedback that the intelligent coaching agents have provided.

References

[1].Yen, J., Yin, J., Ioerger, T.R., Miller, M., Xu, D., and Volz, R.A. *CAST: Collaborative Agents for Simulating Teamwork*. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01)*. 2001.

[2].Kleinman, D.L., Young, P. W., and Higgins, G. *The DDD-III: A Tool for Empirical Research in Adaptive Organizations*. In *Proceedings of the Command and Control Research*. 1996.

[3].Woods, D.D., *Cognitive Techniques: The Design of Joint Human-Machine Cognitive Systems*, in *AI Magazine*. 1986. p. 86-92.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

P^3T: A System to Support Preparing and Performing Peer Tutoring

Emily Ching[1], Chih-Ti Chen[2], Chih-Yueh Chou[3], Yi-Chan Deng[2], Tak-Wai Chan[1]
Graduate Institute of Network Learning Technology[1]
Department of Computer Science & Information Engineering[2]
National Central University, Taiwan
Department of Computer Science and Engineering, Y uan Ze University, Taiwan[3]
smileming@lst.ncu.edu.tw, chihtichen@lst.ncu.edu.tw,
cychou@saturn.yzu.edu.tw, ycdeng@cl.ncu.edu.tw, chan@cl.ncu.edu.tw

Abstract. Peer tutoring has been proven an effective way to engage students in active learning. This paper presents a system called P^3T that supports peer tutoring in a classroom where every student is equipped with TabletPC or laptop with wireless capability. P^3T structures peer tutoring by scaffolding both tutors and tutees to prepare for the tutorial session and facilitating their elaborations during their face-to-face tutoring. The rationales of the evolving design of P^3T prototype are given and discussed in this paper.

1. Introduction

Students teaching students, or peer tutoring, is a pedagogical strategy that has been being studied extensively in education research. It has been found that having students teach each other increases their achievements at various educational levels [1][2]. However, classroom peer tutoring programs without technology support are usually for learning simple tasks. For example, the peer tutor in class-wide peer tutoring program reads the questions and answers on a set of flashcards when interacting with the tutee [3][4]. We assert that when a classroom where every student is able to interact with her classmates via her own computing device with wireless capability unobtrusively, it is probable that we can design more sophisticated support for peer tutoring given the computing affordances offered by technology. This paper intends to give an account of the design rationales of our proposed system, P3T, standing for computer supported Preparing and Performing Peer Tutoring, which supports for a complex peer tutoring model by the combination of Web and wireless technologies.

2. Background

Evidences from empirical researches have indicated that students have greater academic performance when they are studying for teaching others than for taking a test [5]. In a more in depth research, Coleman et al. [6] found that when carrying out far transfer tasks (e.g. inference and application), students who were told to teach others by explanation outperformed those who were told to teach by summarization. This finding suggests that the type of teaching task is sensitive to tutors' learning outcome; in other words, complex teaching tasks will involve student tutors in deeper thinking processes during the preparation for teaching.

Besides preparing for teaching, performing teaching also bring tutors some cognitive benefits. This phase includes verbally presenting instructional materials and responding to tutee's questions. Webb [7] noticed that the former makes learning occur when one is aware of any inadequacies during giving out elaboration, and the latter makes one learn because of further clarification and discovered that students who gained the most from cooperative activities were those who provided elaborated explanations to others.

When comparing tutors' learning outcomes with tutees', however, there are many studies demonstrating that peer tutoring benefits tutors in their cognitive gains more than it benefits tutees [5]. According to Webb's finding [7] as summarized previously, it can imply that if a peer tutoring model has tutors cover most of the elaboration activities and tutees act as passive listeners, it will only favour tutors' learning.

3. Design Rationales

Conclusions drawn from the empirical researches are that the peer tutoring program should take advantage of learning effects during preparation for teaching, especially for teaching tasks involving deep-processing thinking, as well as during performing teaching. Tutees should also be involved in active learning to increase their cognitive gains in a peer tutoring setting. Based on these suggestions, P^3T is featured in three designs of which we give an account in this section.

3.1 Computer-Scaffolded Tutorial Notes Composition

One main feature of P^3T is having each student tutor compose her own teaching material. By composing the tutoring notes, students concretely shape and present their thoughts. And, while developing and revising the tutorial notes, the composer is actually reflecting on her own thoughts and consequently involving her in deeper thinking. By placing what and how to be taught into digital text or figures, tutors make their comprehension of the target material "visible" and hence able to be monitored by the class teacher with the support of P^3T system. However, without guidance, students may not know how to compose a good tutorial note [8]. P^3T system scaffolds tutors to self-test their understanding, make lesson plan, identify keywords and their connectedness with prior knowledge, structure the tutorial notes in general format, and self-assess their tutorial notes with rubrics which assess the quality of tutorial notes from five perspectives: completeness, correctness, reorganization, presentation, and tutor's personal analysis.

3.2 Computer-Supported Collaborative Tutoring

After guiding tutors in individually composing tutorial notes, P^3T further enhances the qualities of tutorial notes and tutoring performance by supporting the collaboration among tutors. The mechanisms are three sequential stages: anonymous peer assessment of tutorial notes, pairing tutors and having each pair generate a common tutorial note, and tutors in a pair helping and consulting each other during the tutorial session.

Tutors assess each other's works according to the same rubrics of the self-assessment. Besides providing the anonymous assessment function, the system distributes the tutorial note which earns the highest score to all tutors for their reference. After peer assessment, tutors are paired to integrate their tutorial notes into a common version. Tutors in a pair further clarify, reflect on, and merge their thoughts while confronting alternative points of view in this integration process.

In the tutorial session, each student tutor has her own peer learner to teach, and the P^3T system presents the common tutorial note for the tutoring dyad. Tutors with common tutorial note help and consult each other when confronting questions from their tutees, and then the four students—two tutoring dyads—discuss together as a '*small learning group*'.

3.3 Computer-Facilitated Tutor-Tutee Face-to-Face Elaboration

In the model of P^3T, tutoring is held in the form of class-wide one-on-one tutoring in a classroom where each student has a TabletPC or laptop equipped with wireless connection capability. Besides displaying tutor-designed tutorial notes and questions for the tutees, P^3T system also prompts questions which tutees posed in their pre-class reading. In other words, tutees have to come to the tutorial session with ideas in mind. Using P^3T system, tutees point out the parts they feel difficult and pose questions about the original learning materials during pre-class reading phase, and the system will prompt those difficulties and questions for tutors to respond to during the tutorial session. Another P^3T's design to facilitate the elaboration and interaction between tutors and tutees is distributing instructor-designed questions for all students—tutors and tutees—to answer individually initially, and then having the four members in a *small learning group* share their answers and discuss for the correct one.

4. Summary and Current Work

The current work of P^3T is an ongoing research effort to construct a model of learning by peer tutoring in 1:1 classroom setting. The underlying framework for the research is aimed at integrating active learning processes (e.g. interpretation, elaboration, organization, etc.) with theories of learning by teaching and cognitive skill development. The educational context involves tutors' learning by composing tutorial notes individually and then collaboratively, and tutees' learning by pre-class study, and followed by face-to-face elaboration. The technology plays the roles of tutoring scaffolder, collaborative learning supporter, and elaboration facilitator. Currently we are implementing this system in a graduate course and will report our findings in the near future.

References

[1] Cohen, P.A.; Kulik, J.A.; Kulik, C.L.C. (1982). Educational outcomes of tutoring: a meta-analysis of findings. *American Educational Research Journal*, *19*, 237-248

[2] Falchikov, N. (2001). *Learning together: Peer tutoring in higher education.* London: RoutledgeFalmer.

[3] Greenwood, C.R., Delquardri, J.C., and Hall, V. (1989). Longitudinal effects of classwide peer tutoring, *Journal of Educational Psychology*, *81*, 371–383

[4] Fantuzzo, J. W., King, J. A., & Heller, L. R. (1992). Effects of reciprocal peer tutoring on mathematics and school adjustment: A component analysis. *Journal of Educational Psychology*, *84*, 331-339.

[5] Bargh, J. A., & Schul, Y. (1980). On the cognitive benefits of teaching. *Journal of Educational Psychology, 72*, 593-604.

[6] Coleman, E. B., Brown, A. L., & Rivkin, I. D. (1997). The effect of instructional explanations on learning from scientific texts. *Journal of the Learning Sciences, 6*, 347-365.

[7] Webb, N. M. (1989). Peer interaction and learning in small groups. *International Journal of Educational Research, 13*, 21-39.

[8] Ching, E., Chen, C. T. , Chou, C. Y., & Deng, Y. C. (2005). A pilot study of computer supported learning by constructing instruction notes and peer expository instruction. Short paper will be presented in the 10th conference of Computer Supported Collaborative Learning (CSCL 2005).

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Cognitive and Motivational Effects of Animated Pedagogical Agent for Learning English as a Second Language

Sunhee CHOI and Hyokyeong LEE
University of Southern California/Information Sciences Institute
Los Angeles, CA 90089, U.S.A

Abstract. This paper discusses the results of a pilot study that explores the cognitive and motivational effects of an animated pedagogical agent as well as an alternative delivery system (a simple flashing arrow with audio) in a multimedia environment in which college level ESL students learn English relative clauses. The study also examines the cognitive efficiency of these two media systems used to deliver the same instructional method for teaching English grammar.

Introduction

The primary purpose of this pilot study is to examine the claims that Animated Pedagogical Agents (referred as a pedagogical agent hereafter) increase learning scores over instructional treatments that do not employ agents [1][3][4]. The present study explored the cognitive and motivation effects of an agent as well as an alternative multimedia system (i.e., simple flashing arrow with audio) in a computer-based learning environment in which college level ESL (English as a Second Language) learners learn English relative clauses. The study also measured the cognitive efficiency of the two multimedia systems. Cognitive efficiency refers to "one medium being more or less effortful than another, more or less likely to succeed with a particular learner, or interacting more or less usefully with a particular prior knowledge set" (p. 25) [2], leading to faster learning, or requiring less conscious effort from learners in processing learning material.

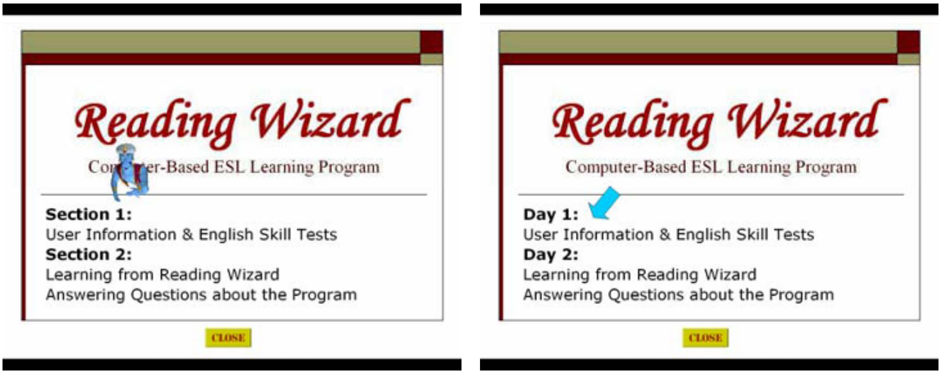
1. Study Design and Method

The design of the present study is a true experimental design with pre- and posttest and involves two treatment groups. An explicit presentation of rule on English relative clauses (e.g., how the target grammar works and what strategies can be used to process it) and a reading comprehension task were adopted as an instructional treatment in the study. Both tasks were computerized and delivered online. An animated pedagogical agent, Genie' was embedded in the agent-based learning environment (Agent Group) to deliver the explicit rule presentation, while an flashing arrow with audio was implemented to do the same thing in the non-agent based environment (Arrow Group) [Figure 1].

19 students who speak English as a second language were recruited from two local universities (8 Korean, 6 Chinese, 3 Thai, & 2 Japanese speakers). Participants were randomly assigned to one of the two treatment groups, Agent Group (9 participants) and Arrow Group (9 participants). The pretest consisted of three testing measures: a sentence combination test (12 questions), an interpretation test (9 questions), and a grammaticality judgment test (12 questions). The posttest was essentially the same as the pretest except that

some lexical items were replaced with other equally difficult items and all the items were presented in a different order. Participants learned English relative clauses using the ESL learning program, 'Reading Wizard' that consisted of the explicit explanation on the target grammar and the reading task. Five dependent variables were measured including mental effort, self-efficacy, time, learner interest, and performance measures.

Figure 1. Agent based and Non-Agent based Learning Environments



Reading Wizard

Co[illegible]er-Based ESL Learning Program

Section 1:
User Information & English Skill Tests
Section 2:
Learning from Reading Wizard
Answering Questions about the Program

CLOSE

Reading Wizard

Computer-Based ESL Learning Program

Day 1:
User Information & English Skill Tests
Day 2:
Learning from Reading Wizard
Answering Questions about the Program

CLOSE

Agent Group **Arrow Group**

2. Results & Discussion

2.1Does the explicit rule presentation have positive effect on learners' learning of English relative clauses?

The results of the Paired-Samples *t* test show that there was no difference between the pretest and the posttest scores [t (17) = -1.021, *p* = 0.321]. However, since the half of the participants had high prior knowledge, another t-test was conducted after dropping those who answered correctly more than 80% of the pretest questions. The results show that the students with low prior knowledge made significant improvements after learning from the program [t (6) = -3.315, *p* = 0.016], which indicates that the explicit rule presentation and reading comprehension task did have positive effect on learners' learning of English relative clauses when the learners did not have much prior knowledge of the target grammar.

2.2 Does the type of medium delivering the same instructional method have a differential effect on learners' learning of English relative clauses?

No significant difference was found between the agent and the arrow group on the pretest [t (16) = 0.383, *p* = 0.707], which means that any difference obtained from the posttest is attributable to the instructional treatment implemented in the study. There was no meaningful difference between the two groups on the posttest either [t (16) = 0.118, *p* = 0.907]. The results of Mann-Whitney test (a non-parametric test), however, showed that a delivery medium did make difference in the low prior knowledge participants – Agent group performed better (z = -1.101, *p* = 0.400), which means that a pedagogical agent might work better with low prior knowledge.

2.3 Does the type of medium delivering the same instructional method have a differential *effect on the time and mental effort spent to achieve the same level of performance?*

No difference was found between the two groups in terms of the amount of time spent by the learners to acquire the target grammar (Agent group: *M* = 12.70 minutes, Arrow group: *M* = 12.52 minutes, t = .111, p = .913). However, the learners in Agent group exerted significantly less mental effort than those in Arrow group (Agent group: *M* = 2.17, Arrow group *M* = 3.44, t = -2.682, p = .016). Given that learners' performances in the two groups were not significantly different, it is reasonable to say that the agent group achieved the same level of performance with less mental effort. In other words, the cognitive efficiency of the animated pedagogical agent was higher than the arrow with audio when they were used to deliver the explicit rule presentation.

2.4 Does the type of medium delivering the same instructional method have a differential effect on the levels of learner interest and motivation?

The results of paired-sample t test show that the overall increase of learner self-efficacy beliefs from pre-treatment to post-treatment was not significant [t (6) = -2.202, *p* = 0.070; z = -1.787, *p* = 0.074]. Yet, the arrow group's scores were higher and increased more (almost to the significant level, t (5) = -2.465, *p* = 0.057; z = -2.141, p = 0.032) than the agent group. On the contrary, the agent group (*M* = 5.25) perceived the learning environment more interesting than its counterpart (*M* = 4.5) [t (5) = 0.624, *p* = 0.560; z = -1.061, *p* = 0.289]. In other words, the increased interest did not have a cause and effect relationship with students' actual learning outcomes.

3. Conclusion and Future Research

The present study shed light on the issue of cognitive efficiency of multimedia which has not been included in the field of instructional technology. As discussed above, the delivery medium did not have significant impact on learning product, but it did on learning process, especially the amount of mental effort exerted by learners. The study also demonstrated that animated pedagogical agents work better with learners with low prior knowledge than with those with high prior knowledge. The study also displayed that the agent group showed higher interest level than the arrow group, but the higher interest was not led to better performance. Yet, it should be noted that the there was no clear pattern observed due to the small sample size. That is, it is required to conduct a larger scale study which in fact is being planned by the researchers.

References

[1] Atkinson, R. K. (2002). Optimizing learning from examples using animated pedagogical agents." *Journal of Educational Psychology, 94*(2): 416-427.

[2] Cobb, T. (1997). Cognitive Efficiency: Toward a revised theory of media. *Educational Technology Research & Development*, *45*(4), 21-35.

[3] Johnson, W. L., Ricke. J. W., & Lester, J. C. (2000). Animated pedagogical agents: Face-to-face interaction in interactive learning environments. *International Journal of Artificial Intelligence in Education*, *11,* 47-78

[4] Lester, J. C., Converse, S. A., Kahler, S. E., Barlow, S. T., Stone, B. A., & Bhogal, R. S. (March, 1997). *The persona effect: Affective impact of animated pedagogical agents*. Computer-Human Interaction: Atlanta.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Added value of a task model and role of metacognition in learning

Noor Christoph [1], Jacobijn Sandberg and Bob Wielinga
University of Amsterdam

Keywords. Constructivism, Knowledge Management, Knowledge retention, Metacognition, Problem solving, Task models

1. Introduction

Constructivist learning environments generally advocate the active acquisition of knowledge and skills, collaboration, and the use of authentic and realistic case material. Also, the constructisivm point out the importance of metacognitive skills in order to monitor and control the learning process. In general, the use of metacognitive skills is positively related to learning success [5]. Games and simulations fit rather well in this paradigm since learners can experiment in a highly realistic environment. Empiric results however, show that learning is problematic in these environments [2]. One of the problems concerns difficulties learners have in regulating their learning behaviour. In this paper the assumption is that constructivist learning environments can be beneficial for learning provided that they contain a task model. A task model is a model that prescribes how to solve a particular problem. The task model contains all elementary executable activitives stemming from the general phases of problem solving, decompiled at the level of cognitive activities. Mettes & Pilot [3] for example developed a task model for problem solving in the field of thermodynamics. It appeared that students working with this model, outperformed students that did not have this model available. From the field of instructional technology the notion of including a task model in a learning environment is seen as a form of instructional support, especially for regulating learning behaviour. An assumption is that metacognitive skills of novice learners are mostly domain independent in nature. In order to become an expert, these skills should be instantiated for the domain at hand. Since in a the task model the general phases of problem solving are applied to a particular domain, the task of students is made easier. They do not have to use their general metacognitive skills and apply these to the domain at hand, since this is done for them. This should lead to improved learning compared to a situation in which no model is available. When no task model is available in the learning environment, students have to fall back on their framework of metacognitive skills. They have to apply these general skills to the domain themselves. If these metacognitive skills are not readily

[1] Correspondence to: Noor Christoph, University of Amsterdam, Matrix I, Kruislaan 419 1098 VA Amsterdam. Tel.: +31 020 888 4671; Fax: +31 020 888 0000; E-mail: noor@science.uva.nl.

available, learning will be problematic. Thus the research question in this paper is: *What is the added value of a task model for learning and what role do metacognitive skills play in this context?* The learning environment suitable for this research, is called KM Quest. KM Quest[1] is a constructivist gaming-simulation environment for the domain of Knowledge Management (KM). The task model in KM Quest concerns the Knowledge Management model (KM model). It prescribes how students should solve knowledge management problems. KM Quest will be played in two conditions: one without the task model (no-model condition) versus the standard environment (model condition). Students are assigned to conditions based on randomization. In a pre-test post-test design, measures of learning and self-reported use of metacognitive skills will be employed. Hypothesis 1 covers the main effect of condition: students in the model condition outperform students in the no-model condition with regard to the acquisition of declarative and procedural knowledge. Hypothesis 2 concerns an interaction effect of condition and metacognition. Players in the no-model condition that score high on use of metacognitive skills reach comparable scores on knowledge tests to players in the model condition. Players in the no-model condition that score low on use of metacognitive skills, perform less on knowledge tests than students in the model condition.

2. Methods

The electronic questionnaire KMQUESTions was used in order to measure the acquisition of declarative and procedural knowledge. KMQUESTions was developed in a previous study and appeared to be sufficiently reliable and valid [10]. The scale Metacognitive self-regulation of the Motivated Strategies for Learning Questionnaire [11] is used in order to measure the self-reported use of metacognitive skills. The reliability of this scale is sufficient [10, 11]. Scores on metacognition were divided in two groupes based on the median, in order to discriminate between students that scored high and low on this variable.

3. Results

Concerning within-subject effects, a main effect for learning was found (F = 72.13, p < 0.01). No interaction effects were found. The hypothised interaction effect of condition and metacognition was not found (F = 0.22, p = 0.64). This indicates that students acquired declarative knowledge regardless of condition and metacognition. A main effect for acquiring procedural knowledge could be reported (F = 38.56, p < 0.01). A main effect was found for condition (F = 9.26), p < 0.01). Students in the model condition outperformed students in the no-model condition. One interaction effect was found, namely between learning success and metacognition (F = 4.66, p < 0.05). Students that scored low on metacognition, showed more learning success in relation to students that scored high on metacognition. No interaction effects between condition and metacognition could be reported (F = 0.10, p = 0.75). A complication was that students in the no-model condi-

[1] KM Quest was developed in the EC project KITS (IST-1999-13078), which consisted of the following partners: University of Twente, The Netherlands; TECNOPOLIS CSATA novus, Italy; Cibit, The Netherlands; EADS, France; ECLO, Belgium and the University of Amsterdam, The Netherlands

tion did not have the KM model at their disposal, therefore they could not answer several items for procedural knowledge correctly. After excluding these items, the main effect of learning did hold (F = 15.69), $p < 0.01$). However, the main effect of condition disappeared. The interaction effect between learning success and metacognition also remained (F = 4.55, $p < 0.05$).

4. Discussion and conclusions

In this study, the objective was to find out what the effect of a task model was on learning, and how the use of metacognitive skills fits in. The results reveal that students acquire declarative and procedural knowledge about the domain Knowledge Management, this replicates findings of an earlier study [10]. The first hypothesis cannot be confirmed. Students in the model condition do not outperform their peers in the no-model conditions. The second hypothesis, namely about the interaction effect of condition and metacognition, cannot be confirmed. Students in the no-model condition that score high on metacognition, do not exceed students in the same condition that score low on metacognition. However, an interaction effect was found between learning *success* in terms of (general) procedural knowledge and metacognition. Students that scored low on metacognition, obtained significantly more learning gain in (general) procedural knowledge than students that scored high on metacognition, regardless of condition. Concluding, the main finding of this study is that especially weaker students in terms of metacognitive skills appear to benefit from KM Quest, regardless whether a model is present. Their learning gain is highest compared to students that are stronger in using metacognitive skills. It is however, not the KM model that they benefit most from, since the addition of this model to the environment does not lead to better learning results. Perhaps the fact that KM Quest is in essence a constructivistic learning environment is the reason why weaker students achieve more learning success. Maybe for this learning environment, one has successfully translated the theoretical principles underpinning the constructivism into specific didactical and pedagogical teaching strategies that lead to advanced self-regulatory behaviour and therefore, better learning, especially for those who need it.

References

[1] T. de Jong, W.R. van Joolingen (1998). Scientific discovery learning with computer simulations of conceptual domains. Review of educational research, 68, 179-202.

[2] C.T.C.W. Mettes, A. Pilot (1980). Over het leren oplossen van natuurwetenschappelijke problemen. Enschede:Onderwijskundig Centrum CDO/AVC TH-Twente, The Netherlands.

[3] N. Christoph, J.A.C. Sandberg, B.J. Wielinga (2004). Measuring learning effects of a gaming-simulation environment for the domain of knowledge management. Proceedings of IADIS CELDA conference, Lisbon, Portugal.

[4] P.R. Pintrich, D.A.F. Smith (1993) Reliability and predictive validity of the motivated strategies for learning questionnaire (MSLQ). Journal of educational psychology, 82, 33-40.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Introducing adaptive assistance in adaptive testing

Ricardo CONEJO, Eduardo GUZMÁN,
José-Luis PÉREZ-DE-LA-CRUZ and Eva MILLÁN
Departamento de Lenguajes y Ciencias de la Computación
Universidad de Málaga, Spain.
{conejo, guzman, perez, eva}@lcc.uma.es

Abstract. In this paper, we discuss the development of a theoretical framework for introducing adaptive presentation in adaptive testing. To this end, a discussion of some aspects concerning the adaptive selection mechanism for hints is presented. Some axioms that hints must fulfil are also determined, providing a hint validation procedure.

1. Introduction

Testing is commonly used in many educational contexts with different purposes: grading, self-assessment, diagnostic assessment, etc. In order to improve the efficiency of the diagnosis process, adaptive testing systems select the best question to be asked next according to relevant characteristics of the examinee. In this way, higher accuracy can be reached with a significant reduction in test length. One of the most commonly used approaches for adaptive testing is *Item Response Theory* (IRT) [1], which assumes that the answer to a question depends on an unknown latent numerical trait θ, which in educational environments corresponds to the knowledge of the subject being tested.

In any adaptive educational system, it is necessary to have accurate estimations of the student's knowledge level in order to take the more suitable instructional action. In this sense, *Computerized Adaptive Tests* (CATs) [2] based on IRT provide a powerful and efficient diagnosis tool. In our group we have used this framework to design and implement SIETTE[1] [3], [4], which is a web-based assessment system that implements CATs based on a discretization of IRT.

There can be little doubt that one of the main contributions to educational psychology in the XX century is Vigotsky's *Zone of Proximal Development* (ZPD) [5]. A short operational definition useful for our purposes is given in [6]: the zone defined by the difference between a person's test performance under two conditions: with or without assistance. Soon after the definition of the ZPD, attempts to apply this concept were made in the context of the administration of tests, typically with the aim to classify students with the goal to allocate them in the more appropriate educational program. But the main goal of the work presented here is different: to build a model that allows the integration of adaptive assistance in the adaptive testing procedure within the SIETTE system.

It is widely accepted that hinting is a general and effective tactic for teaching. In [7] it is shown that human tutors maintain a rough assessment of the student's performance (the trait θ in our approach) in order to select a suitable hint. Many Intelligent Tutoring Systems also give hints to the student, like for example, ANDES [8] and Animalwatch [9].

[1] http://www.lcc.uma.es/SIETTE

In our framework, assistance will be represented by *hints*, h_1, …., h_n that provide different levels of support for each test item. By *adaptive assistance* we mean that the hint to be presented will be selected by the system depending on how far in the ZPD is the item, in such a way that it provides the minimal amount of information so that the student is able to correctly answer such item.

The work presented here aims to extend our previous research [10] on the introduction of hints and feedback in adaptive testing. The main goal is now the definition and evaluation of a theoretical framework for adaptive hinting. This paper addresses the definition of such framework, and is structured as follows: next, we discuss several aspects concerning the introduction of hints in adaptive testing environments and then we present some conclusions and future lines of research.

2. Introducing hints in an Adaptive Testing environment

As aforementioned, SIETTE implements CATs and IRT in a web-based assessment tool. In contrast with traditional IRT, θ is defined as a discrete variable. To introduce hints in this model, let us first define some terms:

- *Item*. We use this term to denote a question or exercise posed to a student. The solution of such task will be provided by answering a multiple choice question, that is the conjunction of a stem and a set of possible answers, where only one is correct.
- A *test* is a sequence of items.
- *Hint*. A hint is an additional piece of information that is presented to the student after posing a question and before he answers it. Hints may provide an explanation of the stem, clues for rejecting one or more answers, indications on how to proceed, etc. Hints can be invoked in two different ways: a) *active* (the examinee asks for it) or b) *passiv*e, (the system decides when to present it).

As an example, consider the following test item and some possible hints:

```
What is the result of the expression: 1/8 + 1/4?

     a) 3/4       b) 3/8      c)2/4       d)2/8

     Hint 1: 1/4 can be also represented as 2/8
     Hint 2: First, find equivalent fractions so they have the same denominator
     Hint 3: d is incorrect
```

For our purposes, a simplifying assumption is that *hints do not modify student's knowledge*. This assumption (that the trait θ remains constant during the test) is usual in adaptive testing, and in this case means that hints do not cause a change in examinee's knowledge but a change on the ICC shape. In this way, the hint brings the question from the ZPD to student's knowledge level. In this sense, the combination of the item plus the hint can be considered as a new item. This new (virtual) item is represented by a new ICC whose parameters can be estimated using the traditional techniques. However, both ICC's are not independent. First, the use of a hint should make the question easier, which can be stated as:

Axiom 1. Given a question q and a hint h, let ICC_q and ICC_{q+h} be the ICCs associated to the question and to the combination question+hint, respectively. Then, $ICC_q(k) \leq ICC_q+_h(k)$.

If the examinee uses a combination of hints, the question should become even easier:

Axiom 2. Given a question q, a set of hints H and a hint h $\not\subseteq$ H, for all knowledge levels k, ICCq+H(k) ≤ICCq+H+{h}(k).

If the parameters for such ICCs have been estimated and the axioms above are not satisfied, it means that the "hint" misleads the student and should be rejected. This simple approach provides with a useful empirical method to validate hints.

In adaptive environments, it makes sense to look for a criterion for adaptively selecting the best hint to be presented. Under the ZPD framework, if the student is not able to solve the item but the item is on his/her ZPD, the best hint to be presented would be the hint that brings the item I from the ZPD to the zone of the student's knowledge. So for example if an item I has three associated hints h_1, h_2 and h_3 at different levels of detail, it means that each hint is suitable for a different part of the ZPD, as represented in Figure 1.

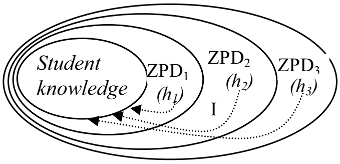


Fig.1. Student knowledge, ZPDs and hints

A possibility for adaptive selection of hints is to use classical adaptive mechanisms: given the knowledge estimation $\theta(k)$ for a student, and given two hints h_1 and h_2, the best hint is the one that minimizes the expected variance of the posterior probability distribution. This mechanism is simple to implement and does not make substantial modifications in the adaptive testing procedure, because the test is used for evaluation and not for learning. However, the use of hints can provide positive stimuli and increase student self-confidence.

3. Conclusions and future work

This paper has presented some ideas about introducing adaptive hints in an adaptive testing environment, based upon IRT constructs. Hints are considered not as knowledge modifiers, but as modifiers of the ICC of a question. Some formal axioms that every model of hints must satisfy have been stated and informally justified. A preliminary evaluation study (not reported here due to lack of space) suggests that that the use of adaptive hints in such environments is adequate and feasible. The next step is the calibration of ICCs for each pair item-hint using empirical data. The obtained ICCs will allow validating such hints and serve as a basis for the integration and implementation of this model in SIETTE to allow for adaptive selection of items and hints in our testing system.

References

[1]. Hambleton, R. K.; Swaminathan, J., and Rogers, H. J. (1991). Fundamentals of Item Response Theory. California, USA: Sage publications.

[2]. H. Wainer (ed.). (1990). Computerized adaptive testing: a primer, Lawrence Erlbaum Associates, Hillsdale.

[3]. Conejo, R., Guzmán, E., Millán, E., Trella, M., Pérez-De-La-Cruz, J. L., and Ríos, A. (2004). SIETTE: A Web-Based Tool for Adaptive Testing. International Journal of Artificial Intelligence in Education, 14, pp. 29-61. IOS Press.

[4]. Guzmán, E. and Conejo, R. (2004). A brief introduction to the new architecture of SIETTE. LNCS 3137, Springer; pp. 405-408.

[5]. Vygotskii, L. S. (1994). The Vygotskii Reader, Blackwell, Cambridge, Massachusetts.

[6]. Wells, G. (1999). Dialogic inquiry: Towards a socio-cultural practice and theory of Education, chapter 10. New York: Cambridge University Press. Online: www.oise.utoronto.ca/~gwells/resources/ZPD.html.

[7]. Gertner, A. S., Conati, C., VanLehn, K. (1998): Procedural Help in ANDES: Generating Hints Using a Bayesian Network Student Model. Proc. 15th National Conference on Artificial Intelligence, pp. 106-111.

[8]. Arroyo, I. (2002). Animalwatch: an arithmetic ITS for elementary and middle school students. Workshop on *Learning Algebra with the Computer* at Intelligent Tutoring Systems 2002, Montreal, Canada.

[9]. Hume, G. D., Michael, J., Rovick, A., Evens, M. W. (1996). Hinting as a tactic in one-on-one tutoring. Journal of Learning Sciences 5(1) pp. 23-47.

[10]. Conejo, R., Guzmán, E. and Pérez-de-la-Cruz, J.L. (2003). Towards a computational theory of learning in an adaptive testing environment. Proceedings of AIED'03, IOS Press: pp.398-400.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Student Questions in a Classroom Evaluation of the ALPS Learning Environment

Albert CORBETT, Angela WAGNER, Chih-yu CHAO, Sharon LESGOLD,
Scott STEVENS, Harry ULRICH
Human-Computer Interaction Institute
Carnegie Mellon University, Pittsburgh, PA 15213 USA

Abstract. Intelligent tutors for problem solving are successful environments, but have limited capabilities to provide help when learning opportunities arise. This paper reports a classroom pilot study of a new ALPS learning environment that is designed to engage students in more active learning by enabling students to ask questions at any time during problem solving. This ALPS environment integrates a Cognitive Tutor for math problem solving with Synthetic Interview technology which allows students to type natural language questions at any time during problem solving and receive pre-recorded videos of a human tutor replying. In this study we examine the rate of question-asking, the content of student questions, and student attitudes about ALPS. The student interactions with ALPS and attitudes toward the environment are promising and provide guidance for future development.

Introduction

Many successful intelligent tutors for problem solving have been developed that lead to demonstrably large achievement gains [1,5]. These tutors can identify learning opportunities during problem solving by detecting student errors, but have a limited capability to help students construct a deep understanding. With rare exceptions [3] students can only press a help key, corresponding to the generic question, “what should I do next.” This paper reports a classroom pilot of a new ALPS (Active Learning in Problem Solving) environment that opens an additional communication channel during problem solving. This environment integrates Cognitive Tutors, a successful problem-solving ITS with an off-the-shelf technology called Synthetic Interviews, to let students ask questions during problem solving. The Synthetic Interview environment [4] permits students to type questions and provides the videotaped replies of a human tutor. A principal goal of providing this virtual human tutor is to engage students more actively in learning, by letting students generate questions at any point during problem solving. The purposes of this classroom study are to observe students’ question-asking behaviors and attitudes toward the new environment.

1. The ALPS Environment

This ALPS environment employs Cognitive Tutor algebra problems. Each problem describes a situation that can be modeled with a linear function. Students solve numerical questions, generate an algebraic expression for the function and graph the function. In this pilot, the ALPS virtual tutor was grafted onto the usual Cognitive Tutor help facilities. Students were advised to start by typing a question in the ALPS window when they needed help. If the video answer to a question (or sequence of questions) was insufficient, students were advised to use the tutor's help button as usual. This ALPS release was seeded with video answers to 70 problem-related questions and 30 responses to social conversational moves (e.g., saying "thanks") or to off-task social questions (e.g., "how old are you").

1.1. A Wizard of Oz Prototype Study

Relatively little is known about student question-asking behavior in problem-solving ITSs. We began studying student question-asking behavior with a Wizard of Oz laboratory pilot of a prototype ALPS environment in which a human tutor played the role of the Synthetic Interview [2]. In this study, students averaged 38.0 utterances per hour. Of these utterances, 14.4 were unprompted questions, 3.5 were questions prompted by the human tutor, 11.8 were other responses to tutor questions or answers, and 8.2 were other comments. We found that more than half the questions were about the interface and a few were requests for definitions. Of the remaining problem-solving questions, shallow answer-oriented questions were far more frequent than deeper process-oriented questions or principle-oriented questions.

2. The Study

One hundred students enrolled in Cognitive Tutor 7th-grade math or pre-algebra courses in two Pittsburgh-area schools participated in the study. Each student used the ALPS Algebra tutor in one Cognitive Tutor class period and completed a short attitude questionnaire.

2.1 Student Utterances

The 100 students generated a total of 548 utterances and averaged 11.1 utterances per hour. This overall rate of utterances is substantially lower than in the Wizard of Oz pilot study, as might be expected in this early pilot, since the ALPS tutor was less proficient at answering questions than the human tutor and the ALPS students had other sources of answers, notably the tutor help button. All student utterances were independently coded by two authors into seven task-related categories and seven off-task social utterance categories. Fifty-three percent of the utterances were task-related and the remaining were social interactions. The distributions of utterances across the task-related categories in this study and in the Wizard of Oz prototype study are essentially the same in their most important feature: among the answer-, process- and principle-oriented mathematics questions, the shallow answer-oriented questions far outnumber those in the deeper categories, and principle-oriented questions are virtually non-existent. This similarity confirms that the scaffolding of deeper knowledge-constructing questions will be a key design goal in an environment that fosters student-initiated questions. Almost half the student utterances were off-task social utterances. Most of these student-initiated utterances were appropriate to and characteristic of interactions with a another human and among these off-task social utterances only 10% were inappropriate, either overly personal or intentionally offensive.

2.2 Questionnaires

Students answered two Likert-scale questions that compared the ALPS and standard cognitive tutors (Which do you like more? Which is more helpful?). Overall, students were neutral between the two tutors. On the 5-point Likert scale, the ratings for the two questions averaged 3.0 and 2.9 respectively. Two other questions asked “What did you like most about the ALPS tutor?” and “What did you dislike most about the ALPS Tutor?”. Their answers suggest that students like the ALPS concept, but were dissatisfied with this initial implementation. Students’ most frequent answers to the first question were that they liked asking questions and liked talking to a “person.” The most frequent answers to the second question were that the answers didn’t help and the tutor was slow.

3. Condusion

The results of this ALPS pilot study are promising. Students interact with the ALPS virtual tutor much as they do with human tutors, in good ways and bad. Students interacted socially with the tutor much as they would with a familiar human tutor, including observing social conventions. If anything, students spent too much time interacting socially with the tutor. The distribution of answer-, process- and principle- oriented task-related questions is also virtually identical in the ALPS and Wizard of Oz studies. This both validates the ALPS environment and poses the greatest design challenge: scaffolding deeper knowledge-building questions. The apparent validity of the ALPS environment implies that it can be a useful research tool to address these issues. Finally, students’ principal complaints that answers were insufficiently helpful and took too long to start up represent implementation difficulties to be overcome, but students report that they like being able to ask questions and like “talking” to the virtual tutor.

Acknowledgements

This research was supported by National Science Foundation Grant EIA0205301 “ITR: Collaborative Research: Putting a Face on Cognitive Tutors: Bringing Active Inquiry into Active Problem Solving.”

References

[1] Anderson, J.R., Corbett, A.T., Koedinger, K.R. and Pelletier, R., 1995. Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences*, *4*, 167-207.

[2] Anthony, A., Corbett, A., Wagner, A., Stevens, S and Koedinger, K. 2004. Student question-asking patterns in an Intelligent Algebra Tutor. In J. C. Lester, R. M. Vicari & F. Paraguacu (Eds.) *Intelligent Tutoring Systems: 7th International Conference, ITS 2004*, 455-467.

[3] Rosé, C.P., Jordan, P., Ringenberg, M., Siler, S., VanLehn, K. and Weinstein, A., 2001. Interactive Conceptual Tutoring in Atlas-Andes. *Proc. Artificial Intelligence in Education 2001*, 256-266.

[4] Stevens, S.M. and Marinelli, D., 1998. Synthetic Interviews: The Art of Creating a ‘Dyad’ Between Humans and Machine-Based Characters. *Proc. IEEE Workshop on Interactive Voice Technology for Telecommunications Applications.*

[5] VanLehn, K., Lynch, C., Taylor, L., Weinstein, A., Shelby, R., Schulze, K., Treacy, D. and Wintersgill, M., 2002. Minimally invasive tutoring of complex physics problem solving. In S. Cerri, G. Gouarderes & F. Paraguacu (Eds.) *Intelligent Tutoring Systems: Sixth International Conference, ITS 2002*, 367-376.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Scrutability as a core interface element

Marek CZARKOWSKI, Judy KAY, Serena POTTS
School of Information Technologies
University of Sydney
Sydney, NSW 2006, Australia

Abstract

This paper describes a new approach to supporting scrutability of adaptation. Because our previous work indicated that people are unfamiliar with the whole notion that they might control the personalisation of an adaptive learning interface, we have experimented with a new approach. We make the scrutability support blatant and presented by default. We report a user trial indicating that although the users were unaccustomed to the notion that they might understand and control personalization, the did succeed in scrutinizing the adaptation.

1. Introduction

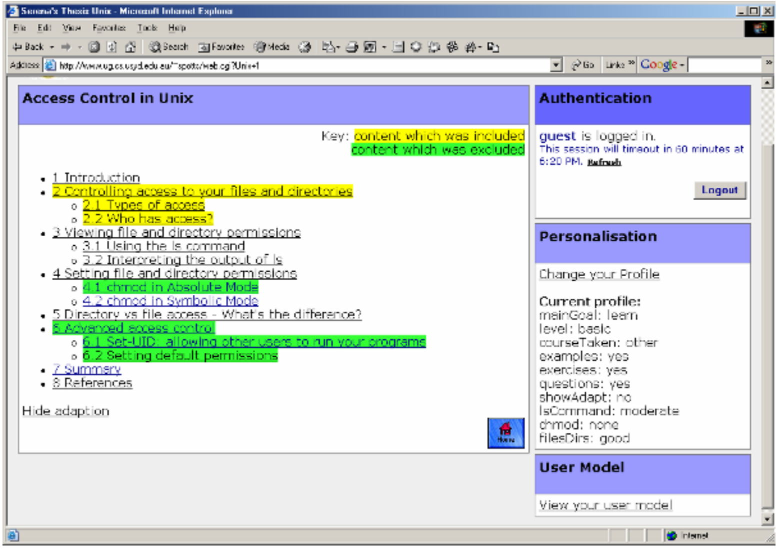
We believe that it is important to be able to build personalized systems in such a way that learners can scrutinize the adaptation of a hypertext system to answer these questions:

- What has been adapted to me?
- What caused the adaptation I saw compared with that seen by a peer?
- How can I control or alter the adaptation?

We have several motivations for this as argued in detail elsewhere [1, 5, 6]. In previous work [1, 5, 6], we have been quite surprised at the difficulty of providing an adaptive hypertext interface that learners are able to scrutinize to answer the questions above. Part of the problem appears to be that people tend to be unaware of the fact that material has been personalized. Even if they realize this, they have difficulty appreciating that the personalization is driven by their student model. Even if they realize this, they have difficulty realizing that they can simply change their user model to effect changes in the personalization. Section 2 presents the user's view of the delivery interface. Section 3 describes the evaluation and Section 4 draws conclusions.

2. System Overview

The system will be described based on a tutorial on UNIX file permissions. Figure 1 shows an example of the interface. Note that the right hand part of the screen is devoted to the personalisation cells. This has the authentication details at the top, then the summary of the user model in text form and finally a cell linked to the student's model. In the figure, the details of the adaptations are displayed. If the user clicks hide adaptation, the screen changes to omit the background colouring and all the content that should be excluded disappears.



Serena's Thesis Unix - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.ug.cs.usyd.edu.au/~spotts/web.cgi?Unix+1
Go Links Google
Internet

Access Control in Unix

Key: content which was included
content which was excluded

- 1 Introduction
- 2 Controlling access to your files and directories
  - 2.1 Types of access
  - 2.2 Who has access?
- 3 Viewing file and directory permissions
  - 3.1 Using the ls command
  - 3.2 Interpreting the output of ls
- 4 Setting file and directory permissions
  - 4.1 chmod in Absolute Mode
  - 4.2 chmod in Symbolic Mode
- 5 Directory vs file access - What's the difference?
- 6 Advanced access control
  - 6.1 Set-UID: allowing other users to run your programs
  - 6.2 Setting default permissions
- 7 Summary
- 8 References

Hide adaption

Authentication

guest is logged in.
This session will timeout in 60 minutes at 6:20 PM. Refresh

Logout

Personalisation

Change your Profile

Current profile:
mainGoal: learn
level: basic
courseTaken: other
examples: yes
exercises: yes
questions: yes
showAdapt: no
lsCommand: moderate
chmod: none
filesDirs: good

User Model

View your user model

Figure 1: Page that has been adapted according to the user's user model showing sections included highlighted in yellow and sections excluded in green

To see why an individual section has been included or excluded, the user holds their mouse is over the section in question and a caption pops up to indicate the reason.

The system has been built using a local lightweight but highly adaptable web framework called Cellerator [2] in conjunction with Personislite [7]. Following the terminology of Brusilovsky [4], the system provides adaptive presentation and adaptive navigation. We have also drawn on many elements of the previous implementation of a scrutably adaptive hypertext [1, 5, 6]. The important difference is that this time we have experimented with making the user model present at all times. This should make the possibility of scrutiny of the personalisation more obvious to users.

3. Evaluation

The system was used to assess whether participants could:

- Appreciate that their profile caused the adaptation;
- Determine what had been adapted to them;
- Understand why it had been adapted and
- Change their user model to control the adaptation.

The evaluation was undertaken in two stages. In first stage, participants 1-5 were asked to answer the initial questionnaire as if they were a single fictitious user, Fred, as in [1]. By contrast, participants 6-9, in the second stage, answered for themselves. Users were required to perform tasks that required them to use the system to scrutinise the adaptation, determine which attributes in their user model caused the adaptation, and finally change the values in their user model to change the adaptation.

All first stage participants understood that their user profile would cause the adaptation of content within the system and were able to effectively change their user profile. Most were able to view the adaptation though of those who could, the majority experienced difficulties utilising the mouseover function provided to see the reason for the adaptation. This is a significant improvement over the previous study [1].

The second stage indicated that in general participants found the system easy to use and were able to use all the functions provided. Only one appeared to experience trouble using the system. This appeared to be due to the fact they chose to see the adaptation on every page when they filled in their user profile initially.

4. Discussion and Conclusions

The purpose of a scrutable adaptive hypertext system is to give the user control, allowing them to understand and control the adaptation. The majority of participants could identify that their profile caused the adaptation, were able to see what had been adapted to them; understood why it had been adapted; could change their user model, hence controlling the adaptation. This is a real step forward, compared with our previous studies [1] with a more subtle interface. This study seems to indicate that if we want learners to scrutinize an adaptive hypertext learning environment, we may need the blatant always-present reminder that adaptation is being performed as we have done in the *Personalisation* cell at the right of the interface.

5. References

[1] Kay, J. and M. Czarkowski, (2003). *How to give the user a sense of control over the personalization of AH?* Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems, User Modeling 2003 Session: p. 121-132.

[2] Kummerfeld, B. and P. Lauder, *Cellerator,* http://www.it.usyd.edu.au/~bob, accessed August 2003

[3] Bicking, I., *The Web Framework Shootout,* http://www.python.org/pycon/papers/framework/web.html, accessed August 2003

[4] Brusilovsky, P., *Adaptive Hypermedia: From Intelligent Tutoring Systems to Web-Based Education.* Intelligent Tutoring Systems 2000, 2000: p. 1-7.

[5] Czarkowski, M., *An adaptive hypertext teaching system.* Honours Thesis, Basser Dept of Computer Science, University of Sydney, 1998.

[6] Czarkowski, M. and J. Kay, *Challenges of Scrutable Adaptivity.* Proceedings of AIED Conference, 11th International Conference on Artificifial Intelligence in Education, IOS Press, 2003: p. 404 - 407.

[7] Kay, J., B. Kummerfeld, and P. Lauder, *Personis: A Server for User Models.* Adaptive Hypermedia and Adaptive Web-Based Systems, AH 2002 Proceedings, 2002: p. 203-212.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

DCE: A One-on-One Digital Classroom Environment

Yi-Chan Deng, Sung-Bin Chang, Ben Chang*, Tak-Wai Chan*
Dept. of Computer Science and Information Engineering, National Central University, Taiwan
*Research Center for Science and Technology of Learning, National Central University, Taiwan
ycdeng@cl.ncu.edu.tw

Abstract. This paper describes a platform that supports for one-on-one digital learning. The platform is named "DCE", which stands for digital classroom environment. It consists of four major modules, curriculum (including activities) module, user data module, communication module, and external interface module. An application example of DCE was described and demonstrated.

1. Background

One-on-one (1:1) digital classroom environment or, simply, 1:1 classroom refers to an environment in classroom setting where every student is equipped with at least one computing device with wireless communication power for individual or group learning, either inside or outside classroom. A group of researchers envision that handy and portable computing devices with wireless support will be accessible by a significant proportion of K12 learners in the forthcoming decade [1].

Researchers have been exploring how to use wireless and mobile devices to enhance the physical classroom. For example, eClass system is one of the earliest digital classroom projects formerly known as Classroom 2000 [2]. ClassTalk [3] focuses on improving teacher ability to pose questions to student's graphic calculators and conduct the class-wide discussion. Gay et al. [4] report on the impact of basic wireless networking in the classroom environment. Roschelle et al. [5] survey a lot of research evidences in the classroom response and communication systems.

DCE 3.0 inherits two previous systems with different names, EduClick and WiTEC. EduClick is another simple wireless response system that can gather collective instant response data from individual students in the class. In experiments conducted with EduClick [6], it was found that there was increase of student interest, attention as well as interactions between students and teachers in the classroom. Instead of using EduClick's simple remote response emitters, WiTEC that built on a revised architecture of EduClick adopted WebPad, the only computing devices that provided handwriting screens of considerable size during the time before TabletPC was first launched in 2002. Small group collaborative learning has been designed in WiTEC and trial tested in elementary classrooms [7]. A particular feature of WiTEC is that it provides a three layer framework to support teachers for designing learning flows [8].

There has been some description of the architectures of EduClick [6] and WiTEC [7]. However, being an inherited system, DCE 3.0 is in an advantageous position for improving, extending and generalizing such 1:1 digital classroom environment in this series of implementation efforts and will be used as a basis for some subsequent experiments [9]. DCE may also bear the potential to impact web-based learning research and development. Currently, hybrid model, that is, distance together with face-to-face classroom instructional model may become the de facto model for network learning. This means that system such as DCE will possibly be an inevitable component of most future learning management systems that have to incorporate both face-to-face and distance learning models. In this paper, we describe the design of the general architecture framework of DCE 3.0, in particular, the four major modules, curriculum (including activities), user data, communication and external interface. An example application of DCE is described.

2. One-on-one Digital Classroom Environment

The general framework of 1:1 digital classroom environment is represented as a communication triangle that plays the orchestrated role (see Figure 1). There are three circles at the three points of the communication triangle. The circle at the upper point is the user circle which includes teacher and students. The outer ring of user circle is the external interface. Teacher and students perform a learning task through the external interface and communication triangle. When students are performing a learning task, the external interface collects user data logs. The circle at the lower left point is the user data circle. User data repository is the source for building student model if the system intends to provide intelligent support for both the students and the teacher. The right hand circle is curriculum which coordinates task and material to build a *curriculum tree* [10] for learning.

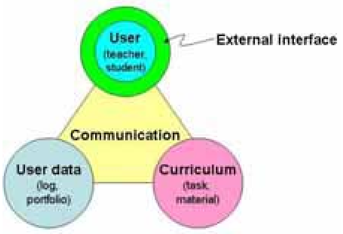


Figure 1 General architecture framework of digital classroom environment

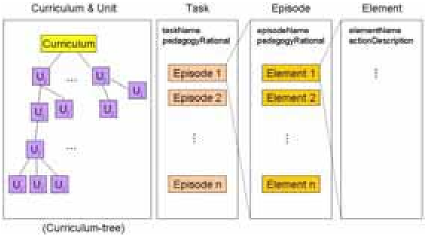


Figure 2 Five levels of curriculum tree

Curriculum module: There are five levels of curriculum module, namely, curriculum level, unit level, task level, episode level, and element level. Figure 2 shows the relation of each level.

Communication module: Students interact in 1:1 classroom is via computer mediated communications. Communication module manages the intra-classroom communication and inter-classroom coordination. The inter-classroom coordination of DCE is prepared for supporting inter-classroom learning activities in the future.

User Data module: User data module is the data centre of student learning records. In the 1:1 classroom, each student is equipped with at least one device, the learning processes can be recorded in student's logs. For example, these logs include the access of curriculum tree, units, tasks, episodes and elements, outcome of tasks, and so on.

External Interface: External interface handles the user interface and peripheral of the learning devices. For example, the peripheral can be digital camera, digital microscope, probes, or other computer embedded tangible objects. It makes the digital learning environment extended easily and naturally. The learning devices also provide the computing affordances to enhance interactions.

3. Application examples

This section describes an example of graduate-level course on "The Trend of Digital Learning Development". Figure 3 shows the unit level curriculum tree of this course. There are 6 units in the tree. The task level description of asking a good question (AGQ) model [9] is presented in the table 1 left column. First is the task name, second is the abstract of this task, third are pedagogy rationales, and last is a set of episode lists. In this case, there are eight episodes of AGQ task. The first argument of composing episode is Q&A which stands for the composing outcome and the

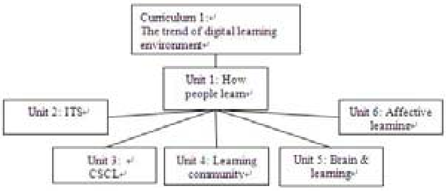

Curriculum 1:
The trend of digital learning environment
Unit 1: How people learn
Unit 2: ITS
Unit 6: Affective learning
Unit 3: CSCL
Unit 4: Learning community
Unit 5: Brain & learning

Figure 3 Curriculum tree of the trend of digital learning development

second argument stands for interactive-mode which, in this case, is an individual action. The episode level description of composing episode is presented in the table 1 right column. This episode includes episode name, general description, pedagogy rationales, and a set of element lists.

Table 1 Levels descriptions of AGQ model

taskName: AGQ *description:* ● Teacher-led presentation of learning material ● Self-study of learning material and individual Q&A generation ● Q&A assessment ● Small group formation and conflict resolution ● Teacher-led class-wide discussion. *pedagogyRationales:* AGQ is a model of student question generation that engaging students in a challenging learning activity that potentially involves higher-level cognitive processing operations. *episodeList:* 1. Reading(Null, Ind, PaperNo) 2. Composing(Q&A, Ind) 3. PeerAssessment(AssSheet, PTraG) 4. Composing(Q&A, TraG) 5. PeerAssessment (AssSheet, G2G) 6. Composing(Q&A, S2S) 7. Quiz(Grade, Ind) 8. Summary(Null, Cls)	*episodeName:* Composing *description:* ● Each student composes a question and a corresponding answer with guided stems and a specific question category ● The question can be multiple choice or short answer open question ● The student assesses his/her own question with a set of rubrics *pedagogyRationales:* ● Questioning can help students find out what the important part in the reading materials is ● The questions will be higher level with the guided stems and the specific question category ● The process of self-assessment can help students reflect on their own questions *elementList:* 1. guidedStem(); 2. composingProduct(); 3. selfAssessment();

4. Summary and future extension

In this paper, we describe the design and implementation of the architecture framework of DCE 3.0, in particular, the four major modules: curriculum module, user data module, communication modules, and external interface module. An example application of DCE is described. The future extension of DCE includes student model of user data module, script engine of communication module, and multi-sensor of external interface. More subsequent experiments using DCE 3.0 platform will be conducted in this coming year.

References

[1] Global network of collaborative researchers on 1:1 educational computing. http://www.g1on1.org, March 2004.

[2] Abowd, G.D. (1999). Classroom 2000: an experiment with the instrumentation of a living educational environment. *IBM Systems Journal*, **38**(4), 508-530.

[3] Dufresne, R.J., Gerace, W.J., Leonard, W.J., Mestre, J.P. & Wenk, L. (1996). Classtalk: A classroom communication system for active learning. *Journal of Computing in Higher Education*, **7,** 3-47.

[4] Gay, G., Stefanone, M., Grace-Martin, M. & Hembrooke, H. (2001). The effects of wireless computing in collaborative learning environments. *International Journal of Human-Computer Interaction*, **13**(2), 257-276.

[5] Roschelle, J., Penuel, W.R. & Abrahamson, L.A. (2004). The networked classroom. *Educational Leadership*, **61**(5), 50-54.

[6] Huang, C.W., Liang, J.K. & Wang, H.Y. (2001). EduClick: A Computer-supported Formative Evaluation System with Wireless Devices in Ordinary Classroom. In Proceedings of ICCE 2001, 1462-1469.

[7] Liu, T.C., Wang, H.Y., Liang, J.K., Chan, T.W., Ko H.W. & Yang, J.C. (2003). Wireless and mobile technologies to enhance teaching and learning. *Journal of Computer Assisted Learning*, **19**(3), 371-382.

[8] Wang, H.Y., Liu, T.C., Chou, C.Y., Liang, J.K., Chan, T.W. & Stephen Yang. (2004). A Framework of Three Learning Activity Levels for Enhancing the Usability and Feasibility of Wireless Learning Environments. *Journal of Educational Computing Research*, **30** (4), 309-329.

[9] Chang, S.B., Tung, K.J., Huang, H.M. & Chan, T.W. (2005). AGQ: A Model of Student Question Generation Supported by One-on-One Educational Computing, In the Proceedings of CSCL2005. (Accepted)

[10] Chen, Y.H., Wu, Y.T., Ku Y.M., Wu J.F. & Chou Y.C. (2005). Curriculum tree: A management system to support curriculum and learning activities, Unpublished paper submitted to the 5th IEEE International Conference on Advanced Learning Technologies (ICALT 2005).

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Contexts in Educational Topic Maps

Christo DICHEV and Darina DICHEVA
Winston-Salem State University,
601 M.L.K. Jr. Dr., Winston Salem, N.C. 27110, USA

Abstract. This paper explores the idea of using contexts to support more efficient information search in Topic Maps-based digital libraries. The notion of context is perceived as abstraction of grouping of domain concepts and resources based on the existing semantic relationships between them. The proposed model of context is used for context representation in the TM4L environment.

Introduction

There is a large amount of high quality learning resources on the web already and they should be made more accessible to users. In this paper we explore the idea of using *contexts* to support more efficient information search. We propose to define contexts as abstraction of clusters of domain concepts and resources based on the existing relationships between them. This is related to our previous work on contexts as well as the development of a framework of concept-based digital course libraries [1]. The framework is based on using the new Semantic Web technology Topic Maps (TM) [2] that enables users to navigate and access documents in an organized manner.

In the topic map paradigm the *scope* feature defines the extent of validity of an assertion: the context in which a topic name or an occurrence is assigned to a given topic, or in which topics are related through associations. Thus thinking of representing contexts in TM, a quick straightforward answer would be to use the topic maps *scoping*. In the TM standard a *scope* is a set of *themes (of validity)*. Themes can be defined and applied to objects (topic names, resources, and associations). Obviously a scope can be used to present a context however this would be a rather static view. Independently of the standard we propose using TM associations to represent *context as grouping*. Topic maps associations can be interpreted as statements relating topics. For instance, in the case of educational applications, it is possible to express the statement that a given concept is represented in a particular learning object (e.g. *tutorial, definition,* etc.) in the form: topic X *is represented by* tutorial Y (in a particular syntactic form). Similarly, associations such as Prolog *is based on* Resolution, Prolog *refers to* Horn-Clause Logic, Prolog *applies* Backtracking, make the topic Prolog pertinent to the related topics. Obviously, association types combined with role types enable meaningful grouping of topics that we call *context.*

Formally context can be defined as a collection of statements that are true in a model. In a less formal perspective, context can be interpreted as the things, which surround, and give meaning to something else. The statement “Snow is white” is meaningful if we talk about New Year in Alaska, but has no meaning in terms of CPU scheduling. We can view contexts as a means of grouping facts relevant to a particular situation. Grouping and classification of objects is a human invention to simplify communication. For our purpose we take a restricted model of this view of context, namely, as a grouping of topics based on their relations to a given topic. Translated in TM terminology a context can be defined as a collection of associations related to a common topic selected to represent and name the context. Technically, this is a nested TM drawn around a topic chosen to name the context.

1. Context as grouping

Most works related to formalizing context are centered around the so called "box model", where "Each box has its own laws and draws a sort of boundary between what is *in* and *out*" [3], [4]. The problem with this approach is that we have to predefine all potentially needed "boxes" in order to use them. The world is too unpredictable to foresee the complete set of contexts that might be needed. Rather than preparing a set of static boxes we suggest to use a TM model that allows shifting boundaries of the context dynamically based on the current topic. The proposed interpretation of context as a collection of topics surrounding a given topic (denoting the context) is intended to localize the search and the inference within an area of *relevant* topics. It allows us to introduce a measure of relevancy. The interpretation of what are the surrounding topics is relative. At one point a topic can be part of the surrounding collection and at another point it can be viewed as surrounded by some other topics giving meaning to it. The relationships are at the heart of semantics, lending meaning to concepts and resources linked to them.

The basic assumptions underlying the proposed contextual framework include:

- Each context is a collection of topics related to a certain topic of the topic map that plays a role of a *focus or center* of the context.
- The central topic is unique and can be used to name the context.
- All semantically related topics identify regions formed by the topics directly or indirectly related to the center of the context.
- The relevance of a topic to the current context is reverse proportional to its distance to the focus of the context.

According to the last assumption the topics of a collection forming a context have no equal status with respect to that context. Their role in the context depends on the *distance* to the central topic. For each topic, the context maps that topic to a collection of topics whose degree of membership to the context depends on their level of relevancy. Among the valuable features of this context model is that it provides a mechanism to refer to the *current context*, and use it to identify an area of interest within the TM. This implies that searching for relevant information can be localized into a specified area of interest.

2. Context: minimal set of generic relations

Learning content typically embodies related topics, hard to be presented into conventional hierarchical structures. Thus we focus on a model for expressing a broader class of relationships on contextual structures. Our idea was to define *a minimal set of generic relations* which cover the needs of the intended applications. The advantage of such an approach is that generic relations subsume particular instances that might be impossible to articulate in specific terms. Our proposed minimal set of generic relations appropriate for e-learning applications is based on guiding principles including: (1) *Simplicity*: simpler is better other things being equal, and (2) *Scope:* a broader scope is better because it subsumes the narrower ones. We propose the following relations:

- *Part-whole* – a transitive relation that characterizes the compositional structure of an object. It is intended to capture in generic sense structural information that subsumes transitive relations of the type X is *part-of* Y, X is *member-of* Y, X is *portion-of* Y, X is *area-of* Y, X is *feature-of* Y, etc.
- *Relevant-to* - represents a family of asymmetric not necessary transitive relations. It is intended to capture in a generic sense asymmetric relations of the type X is *related-to* Y, X is *used-by* Y, X *refers-to* Y, X *points-out-to* Y etc.

- *Similar-to* - describes relations with symmetric roles assigned to the two role players. It is intended to capture in a generic sense symmetric relations of the type "co-refers" (X *is analogous-to* Y, X *co-mentions* Y, X *is-of-the-complexity-level-of* Y, X *is compatible-with* Y, X *is-matching* Y).

We extend this set with the conventional *superclass-subclass*, and *class-instance* relations. The basic intuition is that the five relations *superclass-subclass, class-instance, part-whole, relevant-to* and *similar-to* represent a sufficient basis of generic relations for e-learning applications. They can be used as a generic grouping of concepts and resources that might be difficult to articulate.

The proposed set of relations provides also a strategy for organizing the information. It supports a shared way of grouping topics by standardizing the used set of relations. The intended application of context in our framework includes the following aspects:

- Identifying an area of interest for more reliable and accurate interpretation of search requests.
- Providing a method for ranking the search results by relevance.
- Providing a framework for topic map visualization.

Context has the potential for enhancing the focus and precision of the search. Situating topics contextually provides additional information derivable from the distance between topics. Thus, search results can be listed with decreasing relevance to the search topics.

4. Conclusion

Efficient information retrieval requires information filtering and search adaptation to the user's current needs, interests, knowledge level, etc. The notion of context is very relevant to this issue. In this paper we propose an approach to context modeling and use in topic maps-based educational applications. It is based on the standard Topic Maps support for associations and defines the context as an abstraction of *grouping related information.* This context model provides a mechanism for referring to the *current context*, and using it to identify a current area of interest within the topic map. The latter is useful for localizing a search for relevant information within the current area of interest. We have used the proposed model of context in the design of TM4L, an e-learning environment aimed at supporting the development of efficiently searchable, reusable, and interchangeable discipline-specific repositories of learning objects on the Web [5].

Acknowledgement

This material is based upon work supported partly by the National Science Foundation under Grant No. DUE-0333069 "NSDL: Towards Reusable and Shareable Courseware: Topic Maps-Based Digital Libraries" and DUE-0442702 "CCLI-EMD: Topic Maps-based Courseware to Support Undergraduate Computer Science Courses."

References

1. Dicheva, D., Dichev, C.: A Framework for Concept-Based Digital Course Libraries, *J. of Interactive Learning Research*, 15(4) (2004) 347-364
2. Biezunski, M., Bryan, M., Newcomb, S.: ISO/IEC 13250:2000 Topic Maps: Information Technology, www.y12.doe.gov/sgml/sc34/document/0129.pdf. [Last viewed December 5, 2004].
3. Giunchiglia F.: Contextual reasoning, *Epistemologia, Special issue on I Linguaggi e le Macchine* XVI (1993) 345–364
4. McCarthy, J.: Generality in Artificial Intelligence, *Communications of ACM* 30(12) (1987) 1030–1035
5. The TM4L Project, http://www.wssu.edu/iis/NSDL/index.html

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Analyzing Computer Mediated and Face-to-Face Interactions: Implications for Active Support

Wouter van DIGGELEN, Maarten OVERDIJK and Jerry ANDRIESSEN
Department of Educational Sciences, Utrecht University
Heidelberglaan 1, 3584CS Utrecht, The Netherlands

Abstract: In this paper, we argue that co-constructive activities of learners are not solely confined to the problem at hand or to the process of collaboration, but that they are also directed at the pre-defined structural features of the tool. Although these features are defined in advance, they become meaningful in interaction. Learners re-construct these structures in activity through a process of appropriation during which 'new' structures emerge that guide the collaborative learning activities. It is hypothesized that the learner-external artefact interactions may be an additional source for modelling problem-solving discussions. We will support our line of reasoning with observations of computer mediated interactions from a study that we carried out in a real-life classroom setting.

1. Introduction

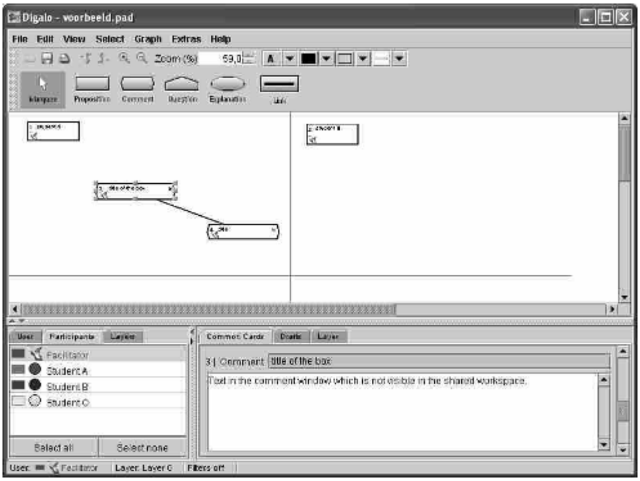
A wide variety of computer tools has been developed to support problem-solving discussions inside and outside the classroom. These tools mediate the interactions between learners and may provide them with active – 'intelligent' – support. This support may take on two forms [1, 3, 5]. First, the computer tools that mediate communication provide the users with pre-defined structures that direct their actions and interactions. This *pre-structured support* concerns, for example, the organisation of the discussion (e.g. turn-taking or simultaneous access), communicative acts (e.g. notation system) or the accessibility of information sources (e.g. anonymity of users). A second kind of – 'intelligent' – support concentrates on the management of collaborative activities. This *active, 'on the spot' support* provides users with current information about their – individual or group – performance which they may use to adjust subsequent actions. This kind of support is based on a system that collects interaction data, transforms that data into *interaction models* that are presented to the learner, directly or after a comparison with a desired state of interaction [3].

A situated perspective on cognition seems promising for developing valid interaction models for active support. The situated approach implies that one should focus on significant features of *learner-environment interaction* that are turned and oriented towards adaptive action [6]. Most interaction models for active support focus on just one part of the learner-environment relation, i.e. the interactions with other learners. We hypothesize that the interaction between the leaner and external artefacts – e.g. the computer tool that mediates their interactions – could be another indicator of effective collaborative problem solving. At least this form of interaction is not as static as is often is assumed.

In line with structuration theory [2] we state that the structural features of the tool are reconstructed in activity through a process of appropriation [4]. In this process of appropriation 'new' structures emerge that guide the collaborative learning activities.

2. Research

We studied the process of appropriation in a real-life classroom experiment where groups of three students argued about a claim. Students could communicate face-to-face *and* with the support of the Digalo tool[1]. The Digalo tool provides its users with a shared workspace based on a concept-mapping interface (figure 1). Users can put forward contributions simultaneously into a shared workspace, using a predefined notation system. Users can also relate associated contributions by using links. When users discuss in the shared workspace they collaboratively construct an – argumentative – diagram of their discussion.



3 | Comment: title of the box

Text in the comment window which is not visible in the shared workspace.

User: Facilitator Layer: Layer 0 Filters off

Figure 1. User interface of the Digalo tool

3. Interactions between the learner and the external artefact

The interactions of learners that are directed towards the external artefact may provide significant input for modelling problem-solving discussions. The learner-external artefact relation can be characterised – just as the learner-learner relation –as mutually constitutive. When learners interact, they appropriate the structural features of the tool. The structures that emerge during their interaction influence subsequent interactions. We studied this process of *appropriation* as it took place in the Digalo environment. We focused our analysis on the structures that users apply to relate their contributions. The structures that emerge when students interact with the tool seem crucial, because they enable the students to construct a visual representation of their discussion. We distinguish three principles for relating and organising contributions in the shared workspace of the Digalo:

1. Users can relate contributions by selecting the *same contribution type* (e.g. 'argument in favour'). These contributions are recognised visually by their shape and colour. This organizing principle is the most compelling because all statements in the Digalo are associated with a contribution type.
2. Users can draw a line – i.e. a *link* – between two associated contributions.
3. Users can *spatially group* associated contributions.

Students could freely apply the three organizing principles. This led to a diversity of argumentative representations. Some groups came up with a rigid, structured representation while other groups constructed a more unstructured, complex map that may even be judged as chaotic by outsiders.

The more rigid, structured maps can be characterised by one leading pattern. However, the leading pattern differed between groups. We observed, for example, maps reflecting the temporal order of the discussion and maps that emphasize the opposing standpoint with regard to the claim. These structured maps are characterized by the fact that all organizing principles strengthen the representation of one leading pattern.

The more unstructured, complex maps lack a clear leading pattern. The students that constructed those maps mainly used the two organizing principles 'same contribution type' and 'link between two associated contributions'. They did not use the third organizing principle – the spatial grouping of contributions – which made these maps more complex. This became even more apparent when the number of contributions increased.

Students' autonomy or freedom of action leads to a diversity in constructed diagrams. Organizing a diagram seems to be a process of – implicit – negotiation that can have multiple outcomes. The time students spent organizing their discussion indicates that the activity of making sense of a discussion is as important as expressing the ideas into words.

A crucial step towards future development of active support lies in examining how individual actions performed in a system relate to social activities, such as argumentation, negotiation and problem solving. Our research implies that the development of interaction models requires a broad analysis of learner-environment interactions in order to understand how individual actions and group interactions constitute the process of argumentation, negotiation or problems solving in computer mediated groups.

[1] The tool described in this paper has been developed in the DUNES project. The DUNES project (IST-2001-34153) is partially funded by the European Commission under the IST Programme of Framework V.

Reference

[1] Barros, B. and Verdejo, M.F. (2000). Analysing student interaction processes in order to improve collaboration: The DEGREE approach. *International Journal of Artificial Intelligence in Education, 11: 221-241.*

[2] Giddens, A. (1986), The Constitution of Society: Outline of the Theory of Structuration. Polity Press, Cambridge.

[3] Jermann, P., Soller, A. and Muehlenbrock, M. (2001). From mirroring to guiding: A review of state of the art technology for supporting collaborative learning. *Proceedings of the first European Conference on Computer-Supported Collaborative Learning,* Maastricht, The Netherlands.

[4] Poole, M.S., Seibold, D.R. and McPhee, R.D. (1996). The Structuration of Group Decision. In R.Y. Hirokwa and M.S. Poole (Eds), *Communication and Group Decision Making*. London: Sage.

[5] Reimann, P. (2003). How to support groups in learning: More than problem solving. *Proceedings of Artificial Intelligence in Education 2003,* Sydney, Australia.

[6] Semin, G.R. and Smith, E.R. (2002). Interfaces of social psychology with situated and embodied cognition. *Cognitive Systems Research, 3: 385-396.*

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Adding a reflective layer to a simulation-based learning environment.

Douglas CHESHER[1], Judy KAY[2], Nicholas JC KING[3]
1. Clinical Biochemistry, Pacific Laboratory Medicine Services, Royal North Shore Hospital,
St Leonards NSW 2065. AUSTRALIA.
Ph: +61 2 9926 5524, Fax: + 61 2 9926 6395
Email: dougc@med.usyd.edu.au
2. Department of Information Technology, University of Sydney
3. Department of Pathology, University of Sydney

Abstract. Computer-based simulations aim to provide an authentic, interactive learning environment. However, there is evidence of failure to acquire deep transferable knowledge. We describe a promising strategy to address this limitation in the form of a *reflective layer* added to the simulation. This paper describes the SIMPRAC reflection model as applied to supporting learning of the management of chronic illness.

Keywords. Medical Education, Computer Simulation, Reflection.

1. Introduction

Medical education is a natural place for simulation-based learning environments. Traditional medical training involves a long period of formal education followed by, or in association with, an apprenticeship, involving practice on human beings (1). This is hampered by the limited time that physicians have for teaching as well as the limited availability of patients as an educational resource (1, 2). After graduation, medical education typically involves a variety of forms such as rounds, educational meetings, conferences, refresher courses, and symposia. Unfortunately, these modalities are frequently ineffective in improving patient care through changes in physician behaviour (3). By contrast, interactive simulation-based learning environments, can effect behavioural change (3). Such simulations can provide opportunities for learning about aspects for which it is not feasible to give students direct experience (1, 2). A variety of simulation environments have been developed with varying levels of fidelity (2, 4-7). Nevertheless, it has been observed that many learners fail to translate experience from a specific simulated case towards broader expertise (7-10). We have been tackling this serious limitation of simulation-based learning environments by building upon the body of evidence that reflection and reflective practice have the potential to improve the effectiveness of learning from experience (11-13). Perhaps surprisingly, its use within medical education has been limited (11). Boyd and Fales observe the importance of reflection for improving the chances of learning from mistakes, rather than continuing to repeat them (14).

In light of these observations, we have defined the SIMPRAC reflective model, which we describe in Section 2. In Section 3, we briefly describe our experiences with it.

2. SIMPRAC model for reflection

SIMPRAC is based upon a basic model for the process of diagnosis and management of chronic illness plus an additional, novel reflective layer. We will briefly describe the basic model and the underlying motivations for its design. Then we will show how the elements of the reflective layer were defined.

The underlying model of the medical practitioners' interaction with the patient is shown in the figure below. It starts with a short introduction to the patient. Then the medical practitioner formulates one or more hypothesis regarding the diagnosis. Further information is obtained iteratively, refining or refuting the hypothesis or diagnosis. Next, the practitioner must decide when they have sufficient information to take a particular course of management . This model can be used for a single consultation or a series of consultations with a number of management cycles, corresponding, for example, to a sequence of appointments, perhaps over many months. Of simulated time. We defined this model to match that taught at our University and constructed the basic simulation environment to match this model.

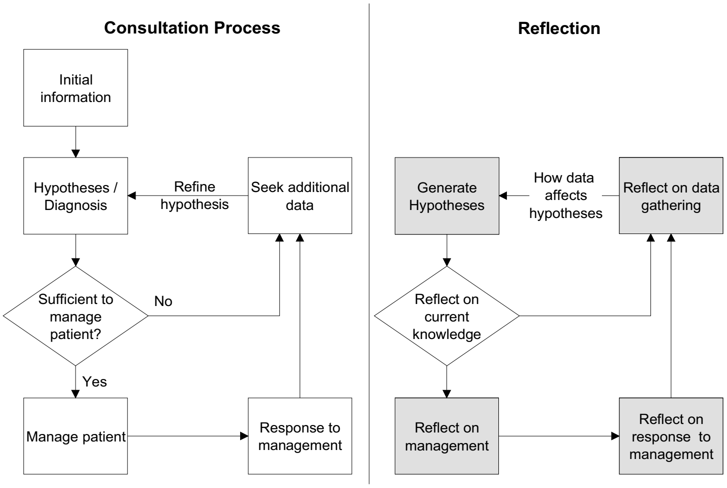


Figure 1: Model of the consultation process. Reflective processes explicitly supported in SIMPRAC are shaded grey.

The novel aspect of SIMPRAC is the reflective layer, as shown at the right of the figure This mirrors the consultation model. Firstly, hypothesis generation is itself a reflective process, in that the leaner needs to reflect on what they know of the patient, and what potential conditions may cause the observed symptoms and signs. From the set of hypotheses, the clinician must then reflect on whether there is enough data, and whether it supports or refutes existing hypotheses, or leads to new ones. If the decision is made that there is insufficient data, further data gathering can be undertaken.

The clinician can then reflect on the appropriateness of this data-gathering, as well as how this process relates to the hypotheses that have been generated. That is, how have the questions, examinations, and investigations contributed to their hypotheses? Are there more effective ways of eliciting the information they desired? Has the way they have asked their questions inhibited or directed the patient's response? How much weight have they given each of these responses when formulating their hypotheses?

Once the clinician decides there is sufficient data to proceed with management, they can then reflect on the various management issues, how these can be best addressed, whether they have sufficient knowledge about best management practices.

Finally, having instituted various treatments and observed the patient's response, the practitioner can again reflect on their management strategies, or such things as the natural history of the disease processes, and how these have influenced the patient's outcome.

With the goal of encouraging reflection and making this reflective process more explicit, we included two activities that interrupt the user's interaction with the virtual patient. These are done at the end of each consultation. We call the first *consultation reflection*. Learners must revisit their actions and reflect on each by assessing its importance, in light of the current situation. The second is the c*omparative performance review*, where learners can see their performance compared with both an expert and the average for their peer group.

3. Discussion

We conducted a qualitative evaluation with ten medical student, five general practitioners, and two specialist medical practitioners. The participants were able to use the reflective elements meaningfully to stand back and reflect on their actions and to self-assess their performance. SIMPRAC enables learners to do reflection both in terms of the precise details of their consultative actions, right down to the point of each individual action. It also supports cohort comparisons. These reflective elements, both the *consultation reflection* on actions during the most recent simulated consultation and the *comparative performance review* which avoids absolute judgement of performance in favour of standards appropriate for the cohort are quite general approaches. The same overall approach could be used as a foundation for a reflective layer for other simulations. With the consultation reflection on actions, the designer of the simulation needs only to identify those actions which the learner should reflect upon, and then the author of the simulation needs to code a set of standard answers for comparison. The comparative performance review is potentially more difficult, as it requires data from the relevant groups of learners. Even this should be a modest additional cost for builders of simulation-based learning environments. Essentially, our reflective layer is a simple but potentially powerful support for improved learning in simulation-based environments.

6. References

[1] Issenberg SB, McGaghie WC, Hart IR, Mayer JW, Felner JM, Petrusa ER, et al. Simulation technology for health care professional skills training and assessment. JAMA 1999;282(9):861-866.

[2] Bergin RA, Fors UGH. Interactive simulated patient-an advanced tool for student activated learning in medicine and healthcare. Computers & Education 2003;40:361-376.

[3] Davis D, O'Brien MAT, Freemantle N, Wolf FM, Mazmanian P, Taylor-Vaisy A. Impact of formal continuing medical education: Do conferences, workshops, rounds and other traditional continuing education activities change physician behaviour or health care outcomes? Journal of the American Medical Association 1999;282(9):867-874.

[4] Hayes KA, Lehmann CU. The interactive patient: a multimedia interactive educational tool on the world wide web. MD Computing 1996;13(4):330-334.

[5] Myers JH, Dorsey K, Benz E. DxR Clinician. Carbondale: DXR Development Group; 2001.

[6] Bloemendaal PM, Eggermont S, Schoonderwaldt EM, Van Baalen JM. Medical training with the dynamic patient simulator. In: Slice of Life; 2002; Toronto; 2002.

[7] Bond WF, Deitrick LM, Arnold DC, Kostenbader M, Barr GC, Kimmel SR, et al. Using simulation to instruct emergency medicine residents in cognitive forcing strategies. Academic Medicine 2004;79(5):438-446.

[8] Elstein AS, Shulman LS, Sprafka SA. Medical problem solving: A ten year retrospective. Evaluation & the Health Professions 1990;13(1):5-36.

[9] Fitzgerald JT, Wolf FM, Davis WK, Barclay ML, Bozynski ME, Chamberlain KR, et al. A preliminary study of the impact of case specificity on computer based assessment of medical student clinical performance. Evaluation & the Health Professions 1994;17(3):307-321.

[10] Eva KW. On the generality of specificity. Medical Education 2003;37:587-588.

[11] Branch WT, Paranjape A. Feedback and reflection: teaching methods for clinical settings. Academic Medicine 2002;77(12 Pt 1):1185-8.

[12] Ruth-Sahd LA. Reflective practice: A critical analysis of data-based studies and implications for nursing education. Journal of Nursing Education 2003;42(11):488-497.

[13] Schön DA. Educating the Reflective Practitioner. Toward a New Design for Teaching and Learning in the Professions. San Francisco: Jossey-Bass; 1987.

[14] Boyd EM, Fales AW. Reflective learning: key to learning from experience. Journal of Humanistic Psychology 1983;23(2):99-117.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Positive and negative verbal feedback for Intelligent Tutoring Systems

Barbara Di Eugenio [a,1] Xin Lu [a] Trina C. Kershaw [a] Andrew Corrigan-Halpern [a] Stellan Ohlsson [a]

[a] *University of Illinois, Chicago, USA*

Abstract. We built three different versions of an ITS on a letter pattern extrapolation task: in one version, students only receive color-coded feedback; in the second, they receive verbal feedback messages when they perform correct actions, and in the third, when they make a mistake. We found that time on task and number of errors are predictive of performance on the post-test rather than the type of feedback.

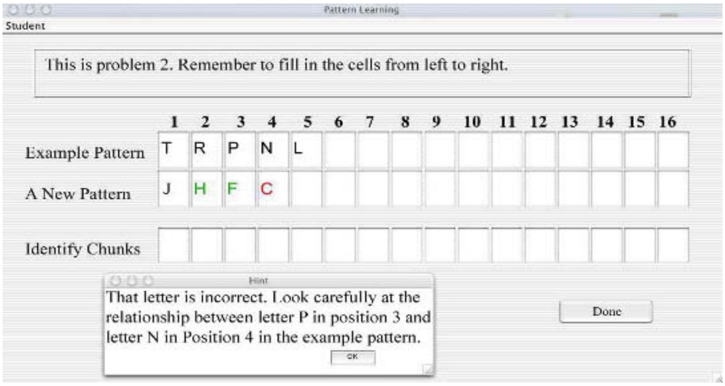
Keywords. Intelligent Tutoring Systems. Natural Language feedback.

1. Introduction and motivation

Research on the next generation of Intelligent Tutoring Systems (ITSs) [2,3,4] explores Natural Language (NL) as one of the keys to bridge the gap between current ITSs and human tutors. In this paper, we describe an experiment that explores the effect of simple verbal feedback that students receive either when they perform a correct step or when they make a mistake. We built three different versions of an ITS that tutors students on extrapolating a complex letter pattern [7], such as inferring MEFMGHM from MABMCDM. In the *neutral* version of the ITS the only feedback students receive is via color coding, green for correct, red for incorrect; in the *positive* version, they receive feedback via the same color coding, and verbal feedback on correct responses only; in the *negative* version, they receive feedback via the same color coding, and verbal feedback on incorrect responses only. In a between-subject experiment we found that, even if students in the verbal conditions do perform slightly better and make fewer mistakes, these differences are not significant. Rather, it is time on task and number of errors that are predictive of performance on the post-test.

This work is motivated by two lines of theoretical inquiry, one on the role of feedback in learning [1], the other, on what distinguishes expert from novice tutors [8]. In another experiment in the letter pattern domain, subjects were individually tutored by three different tutors, one of which had years of experience as a professional tutor. Subjects who were tutored by the expert tutor did significantly better on one of the two problems in the post-test, the more complex one. The content of the verbal messages in our ITSs is based on a preliminary analysis of the language used by the expert tutor.

[1] Correspondence to: B. Di Eugenio, Computer Science (M/C 152), University of Illinois, 851 S. Morgan St., Chicago, IL, 60607, USA. Email: bdieugen@cs.uic.edu.



This is problem 2. Remember to fill in the cells from left to right.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Example Pattern	T	R	P	N	L											
A New Pattern	J	H	F	C												
Identify Chunks																

That letter is incorrect. Look carefully at the relationship between letter P in position 3 and letter N in Position 4 in the example pattern.

Figure 1. The *negative* ITS, that provides verbal feedback on mistakes

2. Method and Results

Our three ITSs are model-tracing tutors, built by means of the Tutoring Development Kit [6]. Fig. 1 shows the interface common to all three ITSs. The *Example Pattern* row presents the pattern that needs to be extrapolated; the *A New Pattern* row is used to enter the answer – the first cell of this row is filled automatically with the letter the extrapolation must start from; the *Identify Chunks* row can be used to identify chunks, as a way of parsing the pattern. If seen in color, Fig. 1 also shows that when the subject inputs a correct letter, it turns green (H, F), and when the subject makes a mistake, the letter turns red (C).

We ran a between-subjects study in which each group of subjects (positive [N = 33], negative [N = 36], and neutral [N = 37]) interacts with one version of the system. All subjects first received instructions about how to interact with the ITS. The positive and negative groups were not informed of the feedback messages they would receive. All subjects trained on the same 13, progressively more difficult, problems, and then received the same post-test consisting of 2 patterns, each 15 letters long. Subjects see the same pattern for 10 trials, but must continue the pattern starting with a different letter each time. Post-test performance is the total number of letters that subjects enter correctly across the 20 trials (a perfect score is 300).

	Post-test score	Time	Errors
Positive	154.06	42.68	18.91
Negative	141.83	45.52	14.69
Neutral	134.62	42.02	21.89

Table 1. Means for the three groups

Means for each condition on post-test scores, time spent in training, and number of errors are shown in Table 1. Subjects in the two verbal conditions did slightly better on the post-test than subjects that did not receive any verbal feedback, and they made fewer mistakes. Further, subjects in the positive condition did slightly better than subjects in the negative condition on the post-test, although subjects in the negative condition made fewer mistakes. However, none of these differences is significant.

A linear regression analysis was performed with post-test scores as the dependent variable and condition, time spent in training, and number of errors as the predictors. The overall model was significant, $R^2 = .16$, $F(3, 102) = 6.52$, $p < .05$. Time spent in training ($\beta = -.24$, $t(104) = -2.51$, $p < .05$) and number of errors ($\beta = -.24$, $t(104) = -2.53$, $p < .05$) were significant predictors, but condition was not a significant predictor ($\beta = -.12$, $t(104) = -2.53$, $p > .05$).

Hence, we can explain variation in the post-test scores via individual factors rather than by feedback condition. The more time spent on training and the higher number of errors, the worse the performance. However, it would be premature to conclude that verbal feedback does not help, since there may be various reasons why it was not effective in our case. First, students may have not really read the feedback, especially in the positive condition in which it may sound repetitive after some training [5]. Second, the feedback may not be sophisticated enough. In the project DIAG-NLP [2] we compared three different versions of an ITS that teaches troubleshooting skills, and found that the version that produces the best language significantly improves learning. The next step in the letter pattern project is indeed to produce more sophisticated language, that will be based on a formal analysis of the dialogues by the expert tutor. On the other hand, it may well be the case that individual differences among subjects are more predictive of performance on this task than type of feedback. We will therefore also explore how to link the student model with the feedback generation module.

Acknowledgments. This work is supported by awards CRB S02 and CRB S03 from UIC, and by grant N00014-00-1-0640 from the Office of Naval Research.

References

[1] A. Corrigan-Halpern and S. Ohlsson. Feedback effects in the acquisition of a hierarchical skill. In *Proceedings of the 24th Annual Conference of the Cognitive Science Society*, 2002.

[2] B. Di Eugenio, D. Fossati, D. Yu, S. Haller, and M. Glass. Natural language generation for intelligent tutoring systems: a case study. In *AIED 2005, the 12th International Conference on Artificial Intelligence in Education*, 2005.

[3] M. W. Evens, J. Spitkovsky, P. Boyle, J. A. Michael, and A. A. Rovick. Synthesizing tutorial dialogues. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*, pages 137–140, 1993.

[4] A.C. Graesser, N. Person, Z. Lu, M.G. Jeon, and B. McDaniel. Learning while holding a conversation with a computer. In L. PytlikZillig, M. Bodvarsson, and R. Brunin, editors, *Technology-based education: Bringing researchers and practitioners together*. Information Age Publishing, 2005.

[5] Trude Heift. Error-specific and individualized feedback in a web-based language tutoring system: Do they read it? *ReCALL Journal*, 13(2):129–142, 2001.

[6] Kenneth R. Koedinger, Vincent Aleven, and Neil T. Heffernan. Toward a rapid development environment for cognitive tutors. In *12th Annual Conference on Behavior Representation in Modeling and Simulation*, 2003.

[7] K. Kotovsky and H. Simon. Empirical tests of a theory of human acquisition of information-processing analysis. *British Journal of Psychology*, 61:243–257, 1973.

[8] S. Ohlsson, B. Di Eugenio, A. Corrigan-Halpern, X. Lu, and M. Glass. Explanatory content and multi-turn dialogues in tutoring. In *25th Annual Conference of the Cognitive Science Society*, 2003.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Domain-Knowledge Manipulation for Dialogue-Adaptive Hinting

Armin Fiedler and Dimitra Tsovaltzi
Department of Computer Science, Saarland University,
P.O. Box 15 11 50, D-66041 Saarbrücken, Germany.

1. Introduction

Empirical evidence has shown that natural language (NL) dialogue capabilities are a crucial factor to making human explanations effective [6]. Moreover, the use of teaching strategies is an important ingredient for intelligent tutoring systems. Such strategies, normally called dialectic or *socratic*, have been demonstrated to be superior to pure explanations, especially regarding their long-term effects [8]. Consequently, an increasing though still limited number of state-of-the-art tutoring systems use NL interaction and automatic teaching strategies, including some notion of hints (e.g., [3,7,5]). On the whole, these models of hints are somehow limited in capturing their various underlying functions explicitly and relating them to the domain knowledge dynamically.

Our approach is oriented towards integrating hinting in NL dialogue systems [11]. We investigate tutoring proofs in mathematics in a system where domain knowledge, dialogue capabilities, and tutorial phenomena can be clearly identified and intertwined for the automation of tutoring [1]. We aim at modelling a socratic teaching strategy, which allows us to manipulate aspects of learning, such as help the student build a deeper understanding of the domain, eliminate cognitive load, promote schema acquisition, and manipulate motivation levels [13,4,12], within NL dialogue interaction. In contrast to most existing tutorial systems, we make use of a specialised domain reasoner [9]. This design enables detailed reasoning about the student's action and elaborate system feedback [2]

Our aim is to dynamically produce hints that fit the needs of the student with regard to the particular proof. Thus, we cannot restrict ourselves to a repertoire of static hints, associating a student answer with a particular response by the system. We developed a multi-dimensional hint taxonomy where each dimension defines a decision point for the associated cognitive function [10]. The domain knowledge can be structured and manipulated for tutoring decision purposes and generation considerations within a tutorial manager. Hint categories abstract from the strict specific domain information and the way it is used in the tutoring, so that it can be replaced for other domains. Thus, the teaching strategy and pedagogical considerations core of the tutorial manager can be retained for different domains. More importantly, the discourse management aspects of the dialogue manager can be independently manipulated.

2. Hint Dimensions

Our hint taxonomy [10] was derived with regard to the underlying function of a hint that can be common for different NL realisations. This function is mainly responsible for the educational effect of hints. To capture all the functions of a hint, which ultimately aim at eliciting the relevant inference step in a given situation, we define four dimensions of hints: The ***domain knowledge*** dimension captures the needs of the domain, distinguishing different anchoring points for skill acquisition in problem solving. The ***inferential***

role dimension captures whether the anchoring points are addressed from the inference per se, or through some control on top of it for conceptual hints. The ***elicitation status*** dimension distinguishes between information being elicited and degrees to which information is provided. The ***problem referential perspective*** dimension distinguishes between views on discovering an inference (i.e., conceptual, functional and pragmatic).

In our domain, we defined the inter-relations between mathematical concepts as well as between concepts and inference rules, which are used in proving [2]. These concepts and relations can be used in tutoring by making the relation of the used concept to the required concept obvious. The student benefits in two ways. First, she obtains a better grasp of the domain for making future reference (implicitly or explicitly) on her own. Second, she is pointed to the correct answer, which she can then derive herself. This derivation process, which we do not track but reinforce, is a strong point of implicit learning, with the main characteristic of being learner-specific by its nature. We call the central concepts which facilitate such learning and the building of schemata around them *anchoring points*. The anchoring points aim at promoting the acquisition of some basic structure, called *schema*, which can be applied to different problem situations [13]. We define the following anchoring points: a *domain relation*, that is, a relation between mathematical concepts; a *domain object*, that is, a mathematical entity, which is in the focus of the current proof step; the *inference rule* that justifies the current proof step; the *substitution* needed to apply the inference rule; the *proof step* as a whole, that is, the premises, the conclusion and the applied inference rule.

3. Structuring the Domain

Our general evaluation of the student input relevant to the task, the *domain contribution*, is defined based on the concept of expected proof steps, that is, valid *proof steps* according to some formal proof. In order to avoid imposing a particular solution and to allow the student to follow her preferred line of reasoning, we use the theorem prover ΩMEGA [9] to test whether the student's contribution matches an expected proof step. Thus, we try to allow for otherwise intractable ways of learning.

By comparing the domain contribution with the expected proof step we first obtain an overall assessment of the student input in terms of generic evaluation categories, such as correct, wrong, and partially correct answers. Second, for the partially correct answers, we track abstractly defined domain knowledge that is useful for tutoring in general and applied in this domain. To this end, we defined a domain ontology of concepts, which can serve as anchoring points for learning proving, or which reinforce the defined anchoring points. Example concepts are the most *relevant concept* for an inference step, that is, the major concept being manipulated, and its *subordinate concept*, that is, the second most relevant concept. Both the domain contribution category and the domain ontology constitute a basis for the choice of the hint category that assists the student at the particular state in the proof and in the tutoring session according to a socratic teaching model [10].

4. Using the Domain Ontology

Structured domain knowledge is crucial for the adaptivity of hinting. The role it plays is twofold. First, it influences the *choice* of the appropriate hint category by a socratic tutoring strategy [2]. Second, it determines the *content* of the hint to be generated.

The input to the socratic algorithm, which chooses the appropriate hint category to be produced, is given by the so-called *hinting session status* (HSS), a collection of parameters that cover the student modelling necessary for our purposes. The HSS is only concerned with the current hinting session but not with inter-session modelling, and thus does not represent if the student recalls any domain knowledge between sessions. Special fields are defined for representing the domain knowledge which is pedagogically useful for inferences on what the domain-related feedback to the student must be. These fields help specify hinting situations, which are used by the socratic algorithm for choosing the appropriate hint category to be produced.

Once the hint category has been chosen, the domain knowledge is used again to instantiate the category yielding a *hint specification*. Each hint category is defined based on generic descriptions of domain objects or relations, that is, the anchoring points. The role of the ontology is to assist the domain knowledge module (where the proof is represented) with the mapping of the generic descriptions on the actual objects or relations that are used in the particular context, that is, in the particular proof and the proof step. For example, to realise a hint that gives away the subordinate concept the generator needs to know what the subordinate concept for the proof step and the inference rule at hand is. This mapping is the first step to the hint specifications necessary. The second step is to specify for every hint category the exact domain information that it needs to mention. This is done by the further inclusion of information that is not the central point of the particular hint, but is needed for its realisation in NL. Such information may be, for instance, the inference rule, its NL name and the formula which represents it, or a new hypothesis needed for the proof step. These are not themselves anchoring points, but specify the anchoring point for the particular domain and the hint category. They thus provide the possibility of a rounded hint realisation with the addition of information of the other aspects of a hint, captured in other dimensions of the hint taxonomy. The final addition of the pedagogically motivated feedback chosen by the tutorial manager via discourse structure and dialogue modelling aspects completes the information needed by the generator.

References

[1] C. Benzmüller *et al.* Tutorial dialogs on mathematical proofs. In *Proceedings IJCAI Workshop on Knowledge Representation and Automated Reasoning for E-Learning Systems*, pp. 12–22, Acapulco, 2003.

[2] A. Fiedler and D. Tsovaltzi. Automating hinting in an intelligent tutorial system. In *Proceedings IJCAI Workshop on Knowledge Representation and Automated Reasoning for E-Learning Systems*, pp. 23–35, Acapulco, 2003.

[3] G. Hume *et al.* Student responses and follow up tutorial tactics in an ITS. In *Proceedings 9th Florida Artificial Intelligence Research Symposium*, pp. 168–172, Key West, FL, 1996.

[4] E. Lim and D. Moore. Problem solving in geometry: Comparing the effects of non-goal specific instruction and conventional worked examples. *Journal of Educational Psychology*, 22(5):591–612, 2002.

[5] N. Matsuda and K. VanLehn. Modelling hinting strategies for geometry theorem proving. In *Proceedings 9th International Conference on User Modeling*, Pittsburgh, PA, 2003.

[6] J. Moore. What makes human explanations effective? In *Proceedings 15th Annual Meeting of the Cognitive Science Society*, Hillsdale, NJ, 1993.

[7] N. Person *et al.* Dialog move generation and conversation management in AutoTutor. In C. Rosé and R. Freedman, eds., *Building Dialog Systems for Tutorial Applications—Papers from the AAAI Fall Symposium*, pp. 45–51, North Falmouth, MA, 2000. AAAI press.

[8] C. Rosé *et al.* A comparative evaluation of socratic versus didactic tutoring. In J. Moore and K. Stenning, eds., *Proceedings 23rd Annual Conference of the Cognitive Science Society*, University of Edinburgh, Scotland, UK, 2001.

[9] J. Siekmann *et al.* Proof development with ΩMEGA. In A. Voronkov, ed., *Automated Deduction — CADE-18*, number 2392 in LNAI, pp. 144–149. Springer, 2002.

[10] D. Tsovaltzi *et al.* A Multi-Dimensional Taxonomy for Automating Hinting. In *Intelligent Tutoring Systems — 6th International Conference, ITS 2004*, LNCS. Springer, 2004.

[11] D. Tsovaltzi and E. Karagjosova. A dialogue move taxonomy for tutorial dialogues. In *Proceedings 5th SIGdial Workshop on Discourse and Dialogue*, Boston, USA, 2004.

[12] B. Weiner. *Human Motivation: metaphor, thoeries, and research.* Sage Publications, 1992.

[13] B. Wilson and P. Cole. Cognitive teaching models. In D. Jonassen, ed., *Handbook of Research for educational communications and technology*. MacMillan, 1996.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

How to Qualitatively + Quantitatively Assess Concepts Maps: the case of COMPASS

Evangelia GOULI, Agoritsa GOGOULOU, Kyparisia PAPANIKOLAOY and Maria GRIGORIADOU
Department of Informatics & Telecommunications, University of Athens, Panepistimiopolis, Ilissia, Athens 15784, Greece
lilag@di.uoa.gr, rgog@di.uoa.gr, spap@di.uoa.gr, gregor@di.uoa.gr

Abstract. This paper presents a scheme for the quantitative and qualitative assessment of concept maps in the context of a web-based adaptive concept map assessment tool, referred to as COMPASS. The propositions are characterized qualitatively based on specific criteria and on the error(s) that may be identified. The quantitative assessment depends on the weights assigned to the concepts/propositions and the error categories.

Introduction

In educational settings, where assessment is aligned with instruction, concept maps are considered to be a valuable tool of an assessment toolbox, as they provide an explicit and overt representation of learners' knowledge structure and promote meaningful learning [6]. A concept map is comprised of *nodes*, which represent concepts, and *links*, annotated with labels, which represent relationships between concepts. The triple *Concept-Relationship-Concept* constitutes a *proposition*, which is the fundamental unit of the map.

The assessment of a concept map is usually accomplished by comparing the learner's map with the expert one [7]. Two most commonly investigated assessment methods are the *structural* method [6], which provides a quantitative assessment of the map, taking into account only the valid components, and the *relational* method, which focuses on the accuracy of each proposition. Most of the assessment schemes proposed in literature either have been applied to studies where the evaluation of concept maps is human-based [7], [5] or constitute a theoretical framework [4], while the number of systems that have embedded a scheme for automated assessment and for feedback provision is minimal [1].

In this context, we propose an assessment scheme for both the qualitative and quantitative assessment of concept maps and subsequently for the qualitative and quantitative estimation of learner's knowledge. The assessment scheme has been embedded in COMPASS (COncept MaP ASSessment tool) (http://hermes.di.uoa.gr:8080/compass), an adaptive web-based concept map assessment tool [3], which serves the assessment and the learning processes by employing a variety of activities and providing different informative, tutoring and reflective feedback components, tailored to learners' individual characteristics and needs.

1. The Assessment Scheme embedded in COMPASS

The proposed scheme is based on the relational method and takes into account both the presented concepts on learner's map and their corresponding relationship(s) as well as the missing ones, with respect to the expected propositions presented on expert map. The propositions are assessed according to specific criteria concerning completeness, accuracy, superfluity, missing out and non-recognizability. More specifically, a proposition is qualitative characterized [3] as (i) *complete-accurate*: when it is the expected one, (ii) *incomplete*: when, at least, one of the expected components (i.e. the involved concepts and their relationship(s)) is incomplete or missing; the error categories that may be identified are incomplete relationship (IR), missing relationship (MR), missing concept and its relationship(s) (MCR) and missing concept belonging to a group and its relationship(s) (MCGR), (iii) *inaccurate*: when, at least, one component/characteristic of the proposition is inaccurate; the error categories that may be identified are incorrect concept (IC), incorrect relationship (INR), concept at different place (CDP) and difference in arrow's direction (DAD), (iv) *inaccurate-superfluous*: when, at least, one component of the proposition is characterized as superfluous; the error categories that may be identified are superfluous relationship (SR) and superfluous concept and its relationship(s) (SCR), (v) *missing*: when the expected proposition is missing (i.e. missing proposition (MP) error), and (vi) *non-recognizable*: when it is not possible to assess the proposition, due to a non-recognizable concept (NRC) and/or a non-recognizable relationship (NRR).

The qualitative assessment is based on the aforementioned qualitative analysis of the errors and aims to contribute to the qualitative diagnosis of learner's knowledge, identifying learner's *incomplete understanding/beliefs* (the errors "MCR", "IR", "MR", CDP", "MCGR", and "MP" are identified) and *false beliefs* (the errors "SCR", "INR", "IC", "SR", "DAD" are identified). The quantitative analysis is based on the weights assigned to each error category as well as to each concept and proposition that appear on expert map. The weights are assigned by the teacher and reflect the degree of importance of the concepts and propositions as well as of the error categories, with respect to the learning outcomes addressed by the activity. The assessment process consists of the following steps (a detailed description is given in [3]):

- at first, the weights of the concepts, that exist in both maps (learner's and expert) and they are at the correct position, as well as the weights of the propositions on learner's map, which are characterized as complete-accurate, are added to the total score,
- for all the propositions/concepts, which are partially correct (i.e. errors "IR", "IC", "INR", "CDP", and "DAD"), their weights are partially added to the total score; they are adjusted according to the weights of the corresponding error categories and added to the total score,
- for all the propositions/concepts, which are superfluous or missing (i.e. errors "SCR", "SR", "MR", "MCR", and "MCGR"), their weights are ignored and the weights of the related concepts, which have been fully added to the score at the first step, are adjusted according to the weights of the corresponding error categories and subtracted from the total score,
- the total learner's score is divided by the expert's score (weights of all the concepts and propositions, presented on expert map, are added) to produce a ratio as a similarity index.

The results of the quantitative and the qualitative assessment are exploited for the provision of adequate personalised feedback according to the underlying error(s) identified, aiming to stimulate learners to reflect on their beliefs.

2. Empirical Evaluation

During the formative evaluation of COMPASS, an empirical study was conducted, aiming to investigate the validity of the proposed scheme, as far as the quantitative estimation of learners' knowledge is concerned. In particular, we investigated the correlation of the quantitative results obtained from COMPASS with the results derived from two other approaches: (i) the holistic assessment of concept maps by a teacher who assigned a score on a scale from 1 to 10, and (ii) the assessment of maps based on the similarity index algorithm of Goldsmith et al. [2]. The study took place during the school year 2004-2005, in the context of a course on Informatics at a high school. Sixteen students participated in the study. The students were asked to use COMPASS and work on a "concept-relationship list construction" task, concerning the central concept of "Control Structures". The results from the assessment of students' concept maps, according to the three different approaches, are presented in Figure 1. The reader may notice that the quantitative scores obtained from COMPASS converge in a high degree with the scores obtained from the other two assessment approaches.

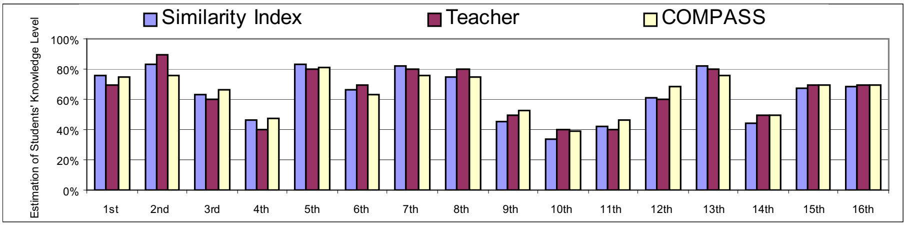


Figure 1. The results of the quantitative assessment of students' concept maps.

3. Conclusions

The discriminative characteristics of the proposed scheme are: (i) the qualitative characterization of the propositions, (ii) the assessment process followed, which takes into account not only the complete-accurate propositions but also the identified errors, (iii) the qualitative diagnosis of learner's knowledge, based on the qualitative analysis of the errors identified, (iv) the quantitative estimation of learner's knowledge level, based on the complete-accurate propositions, on the weights assigned to the concepts, the propositions and the error categories, and (vi) the flexibility provided to the teacher in order to experiment with different weights and to personalize the assessment process. The validity of the proposed assessment scheme can be characterized as satisfactory, as the quantitative estimation of learner's knowledge obtained from COMPASS are close with the estimation obtained from the human-based assessment and the similarity index algorithm.

References

[1] Conlon, T. (2004). "Please argue, I could be wrong": A Reasonable Fallible Analyser for Student Concept Maps. Proceedings of ED-MEDIA 2004, World Conference on Educational Multimedia, Hypermedia and Telecommunications, Volume 2004, Issue 1, 1299-1306.

[2] Goldsmith, T., Johnson, P. & Acton, W. (1991). Assessing structural knowledge. Journal of Educational Psychology, 83, 88-96.

[3] Gouli, E., Gogoulou, A., Papanikolaou, K., & Grigoriadou, M. (2005). Evaluating Learner's Knowledge level on Concept Mapping Tasks. In Proceedings of the 5th IEEE International Conference on Advanced Learning Technologies (ICALT 2005) (to appear).

[4] Lin, S-C., Chang, K-E., Sung, Y-T., & Chen, G-D. (2002). A new structural knowledge assessment based on weighted concept maps. Proceedings of the International Conference on Computers in Education (ICCE'02), 1, 679-680.

[5] Nicoll, G., Francisco, J., & Nakhleh, M. (2001). A three-tier system for assessing concept map links: a methodological study. International Journal of Science Education, 23, 8, 863-875.

[6] Novak, J., & Gowin, D. (1984). Learning How to Learn. New York: Cambridge University Press.

[7] Ruiz-Primo, M., & Shavelson, R. (1996). Problems and issues in the use of concept maps in science assessment. Journal of Research in Science Teaching, 33 (6), 569-600.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Describing Learner Support: An adaptation of IMS-LD Educational Modelling Language

Patricia GOUNON*, Pascal LEROUX** and Xavier DUBOURG*
Laboratoire d'Informatique de l'Université du Maine - CNRS FRE 2730
** I.U.T. de Laval – Département Services et Réseaux de Communication*
52, rue des docteurs Calmette et Guérin
53020 LAVAL Cedex 9, France
{patricia.gounon; xavier.dubourg}@univ-lemans.fr
phone: (33) 2 43 59 49 23

*** Institut d'Informatique Claude Chappe*
Avenue René Laennec
72085 Le Mans Cedex 9, France
pascal.leroux@lium.univ-lemans.fr
phone: (33) 2 43 83 38 53

Abstract. In this paper, we propose an adaptation to the educational modelling language IMS-Learning Design in terms of support activity description and the specification of the actors' roles in these activities. The propositions are based on an organization tutoring model that we have defined. This model has three goals: (1) to organize tasks between actors tutor with learners during a learning session, (2) to allow an adaptive support activity to learners in according to the learning situation and (3) to specify support activity tools of learning environment.

Keywords: tutoring model, educational modelling language, IMS-Learning Design, learner support.

1. Introduction

Different learner support problems are observed in distance learning environments from both the learner and human tutor point of view. A learner may have difficulties concerning in knowing when and about what he could contact the tutor during a learning session. What's more, the learner is not always aware of the mistakes he makes. Therefore, he does not necessarily take the initiative to ask for help. The human tutor may find it difficulty to following the learning activity development. These obstacles affect the human tutor's capacity to react in time and with a suitable learner adapted activity.

These observations give rise to the question: how can we facilitate the design of the accompanying learner environments in the case of distance learning? One response is to offer to guide the designer in the description of the pedagogical scenario of a study unit integrating, in the design process, the learners' planned support.

Presently, the pedagogical scenario descriptions use an Educational Modelling Language (EML). An EML is a semantic model describing the content and the process of a study unit whilst allowing reuse and interoperability [4]. The learner support notion is not often taken into account. It is the reason why we propose an adaptation concerning the EML IMS-LD. The proposition is based on the tutoring organization model that we describe in the next part. We will conclude by giving some perspectives for our research.

2. Model to Organize Tutoring for Learning Activities

Our tutoring model [2] is organized around three components: the tutor, the tutored person and the tutoring style. The tutor component identifies which actor should intervene during the learning activity. The tutored person component defines the beneficiaries of tutor interventions during the learning session. The tutoring style component clarifies the tutoring strategy and the associated tools for actors of learning sessions. To describe the tutoring style, we have to determine (1) the intervention content brought to one or several learners (2) the intervention mode, and (3) actions scheduling. We define four tutoring contents including motivation, which corresponds to a social aspect of tutoring. From this model, the designer describes tutor tasks during the session. Each task identifies the tutor, the beneficiary and the task style. Then, we use each described task to specify tools to support the proposed tutor actions during a learning activity.

The tutoring model is used during the four phases of the life-cycle courseware: design, production, development and evaluation (see Figure 1). The tutoring model application in the life-cycle courseware aims, both to define and understand the tutoring activity better and to facilitate the analysis of the observed tutoring at the end of the learning activity.

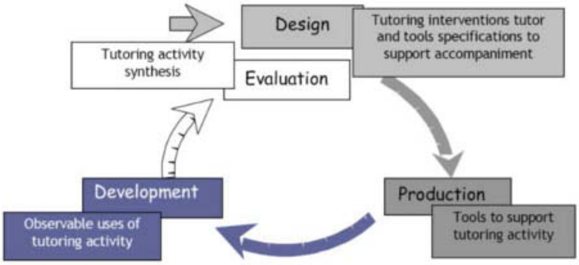

Design
Tutoring interventions tutor and tools specifications to support accompaniment
Tutoring activity synthesis
Evaluation
Development
Observable uses of tutoring activity
Production
Tools to support tutoring activity

Figure 1. Life-cycle Courseware

2. Describing Support Actors Using Norms

There has been a real interest over recent years for the use and application of standards so as to encourage the exchange and reuse of learning objects. [3] defines a learning object as 'any digital resource, used to create learning activities or to support learning and that could be used, re-used or referenced during a learning activity. Different approaches exist to describe learning objects: the documentation approach (LOM) [1], the engineering of software components (SCORM) [5] and the pedagogical engineering (EML).

Our goal is to define what exactly concerns the learner support in pedagogical scenarios. Consequently, it is important to examine how the support is dealt with in the EMLs. We use and make propositions particularly with the language IMS-LD (Open University of the Netherlands) [3]. We choose this language because it allows to model all pedagogical situations and it is opened to modifications. This is important if we want to integrate our tutoring model elements.

This language allows us to describe the development of a study unit using an important diversity of existing pedagogical approaches (constructivism approach, socio-constructivism, …). It use permits us to consider the association of the different contents (pedagogical resources, tools) of a learning design. It also aims to describe the support activity for a unit of study. The description of support activity with IMS-LD is poor and do not allows a precise tutor tasks (tutoring mode, tutoring style, content). The interest of our work consists to add several information allowing to have a better tutoring description for a study unit. To do that, we use the characteristics of the tutoring model.

3. IMS-LD Adaptation Proposal integrating the Tutoring Organization Model

First, the modifications brought to the role component concern the learner and staff components. We add categories (sub-group, co-learner, …) identified in the tutoring model proposed. Thus the granularity for the actors description of a given study unit is increased.

Second, we have added further modifications to the service description. Various information are inserted in the part *<imsld: roles>* to establish the actor references using the tool and the intervention mode used. The aim of the extension proposition is to facilitate the use analysis of the different support tools in a study unit. It is also a way to give better access to tools during the learning activity design.

Third, with the tutoring model, we define a unit of tutor tasks that can be carried out during a learning activity. These tasks help to identify the characteristics and to specify the tool management of the learners' support activity. The tool choice is expressed with IMS-LD. It describes a tutor action by using the tag *<imsld: support-activity>*. The staff references are modified by specifying the characteristics of each actor (tutor and tutored person) and of the exchange style. This description corresponds to the tutor task transcription described in the tag *<imsld: title>*. Then, the task application is defined by one of the tags:

- the task is universal to the study unit (*<imsld:learning-design-ref ref=""/>*),
- the task is specific to a structure activity (*<imsld:structure-activity-ref ref=""/>*)
- or the task described is specific to a learning activity (*<imsld:learning-activity-ref ref=""/>*).

Finally, the tool satisfying the task described is referenced in the tag environment.

4. Conclusion

We proposed, in this paper, an extension to the EML IMS-LD integrating a tutoring organization model that we use to guide the design of support environments. Our proposition aim to add a level of detail to the participating tutor and tutored person's description. This adaptation also brings the same degree of precision to the tool description. Our proposition is used in the environment to guide the designer in the description of the study unit and the specifications of the learner support. The application helps the designer to specify the tool choice for the support activity by proposing a uniform range of tools according to the defined tasks. We also wish to enable the integration of tools and pedagogical scenarios to existing platforms described with IMS-LD.

References

[1] Forte, E. Haenni, F. Warkentyne K., Duval, E. Cardinaels, K. Vervaet, E. Hendrikx, K. Wentland Forte, M. Simillion, F. « Semantic and Pedagogic Interoperability Mechanisms in the ARIADNE Educational Repository », in ACM SIGMOD, Vol. 28, No. 1, March 1999.

[2] Gounon, P., Leroux, P. & Dubourg, X., « Proposition d'un modèle de tutorat pour la conception de dispositifs d'accompagnement en formation en ligne » (à paraître), In: Revue internationale des technologies en pédagogie universitaire, numéro spécial: L'ingénierie pédagogique à l'heure des TIC, printemps 2005.

[3] Koper, R., Olivier, B. & Anderson T., eds., IMS Learning Design Information Model, IMS Global Learning Consortium, Inc., version 1.0, 20/01/2003.

[4] Rawlings, A ; Rosmalen, P., Koper, R., (OUNL), Rodríguez-Artacho, M., (UNED), Lefrere, P., (UKOU), « Survey of Educational Modelling Languages (EMLs) », 2002.

[5] ADL/SCORM, ADL Sharable Content Object Reference Model Version 1.3, Working draft 0.9, 2002.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Developing a Bayes-net based student model for an External Representation Selection Tutor

Beate Grawemeyer and Richard Cox
Representation & Cognition Group
Department of Informatics, University of Sussex, Falmer, Brighton BN1 9QH, UK

Abstract. This paper describes the process by which we are constructing an intelligent tutoring system (ERST) designed to improve learners' external representation (ER) selection accuracy on a range of database query tasks. This paper describes how ERST's student model is being constructed - it is a Bayesian network seeded with data from experimental studies. The studies examined the effects of students' background knowledge-of-external representations (KER) upon performance and their preferences for particular information display forms across a range of database query types.

Keywords. Student modeling, External representations, Bayesian networks

1. Introduction

Successful use of external representations (ERs) depends upon the skillful matching of a particular representation with the demands of the task. Good ER selection requires, *inter alia*, knowledge of a range of ERs in terms of a) their semantic properties (e.g. *expressiveness*), b) their functional roles (e.g. [4],[1]) together with information about the 'applicability conditions' under which a representation is suitable for use [7].

Our aim is to build ERST - an ER selection tutor. We conducted a series of empirical studies (e.g. [6]), that have provided data for ERST's student model and it's adaptation mechanism. This paper extends the work by investigating the effect of learners' background knowledge of ERs (KER) upon information display selection across a range of tasks that differ in their representation-specificity. In the experiments, a prototype automatic information visualization engine (AIVE) was used to present a series of questions about information in a database. Participants were asked to make judgments and comparisons between cars and car features. Each participant responded to 30 questions, of which there were 6 types, e.g. identify; correlate; quantifier-set; locate; cluster; compare negative. Participants were informed that to help them answer the questions, the system would supply the needed data from the database. AIVE then offered participants a choice of representations of the data. They could choose between various types of ERs, e.g. set diagram, scatter plot, bar chart, sector graph, pie chart and table. The ER options were presented as an array of buttons each with an icon depicting, in stylized form, an ER type (bar chart, scatter plot, pie chart, etc). When the participant made his or her choice, AIVE then instantiated the chosen representational form with the data needed to answer the task and displayed a well-formed, full-screen ER from which the participant could read-off the information needed to answer the question. Having read-off the information, subjects indicated their response via on-screen button selections (i.e. selecting one option out of a set of possible options). Note that each of the 30 questions could (potentially) be answered with any of the ER display types offered. However, each question type had an 'optimal' ER. Following a completed response, the participant was presented with the next question in the series of 30 and the sequence was repeated. The data recorded were: the randomized position of each representation icon from trial to trial; user's representation choices (DSA); time to read question and select representation (DSL); time to answer the question (DBQL); responses to questions (DBQA). Further details about the experimental procedure are provided in [6].

Prior to the database query tasks, participants were provided with 4 different types of KER pre-tests [5]. These tests consisted of a series of cognitive tasks designed to assess ER knowledge representation at the perceptual, semantic and output levels of the cognitive system. A large corpus of external representations (ERs) was used as stimuli. The corpus contains 112 ER examples. The decision task (ERD) was a visual recognition task requiring real/fake decisions[1]. The categorisation task (ERC) assessed semantic knowledge of ERs - subjects categorised each representation as 'graph or chart', or 'icon/logo', 'map', *etc.* In the functional knowledge task (ERF), subjects were asked *'What is this ER's function'?*. In the naming task (ERN), for each ER, subjects chose a name from a list. E.g.: 'venn diagram', 'timetable', 'scatterplot', 'Gantt chart', 'entity relation (ER) diagram', *etc* [5].

2. Results and Discussion

The simple bivariate correlations between KER and AIVE tasks for display selection accuracy (DSA), database query answering accuracy (DBQA), display selection latency (DSL) and database query answering latency (DBQL) were: Three of the 4 KER tasks correlated significantly and positively with DBQA (ERD r=.46, p<.05; ERC r=.60, p<.01; ERF r=.66, p<.01); Two KER tasks correlated significantly and positively with DSA (ERC r=.57, p<.01; ERF r=.57, p<.01); DBQA correlated significantly and positively with DSA (r=.30 , p<.01); There is a significant negative correlation between DBQA and DBQL (r=-.28, p<.01); DSA is significantly negatively correlated with DSL (r=.-17, p<.01); There is a significant negative correlation between DSA and DBQL (r=-.32, p<.01); DSL and DBQL are significantly positively correlated (r=.30, p<.01).

The results showed that task performance on three of the KER tasks are better predictors of DBQA performance than DSA. The selection latency results show that a speedy selection of a display type in AIVE is associated with a good display-type choice. This implies that students either recognise the 'right' representation and proceed with the task or they procrastinate and hesitate because of uncertainty about which display form to choose. Less time spent responding to the database query question is associated with a good display-type choice and correct query response. This suggests that the selection and database query latencies may be used in ERST's student model as predictors of students' ER expertise.

[1] Some items in the corpus are invented or chimeric ERs.

Using the experimental data, a Bayesian network [8] was constructed for ERST's student model. Bayesian networks have been applied successfully in ITS (e.g. [2]) and are suitable for recognizing and responding to individual users, and they can adapt to temporal changes. The network will monitor and predict users' ER selection preference patterns within and across query types. It will relate query response accuracy and latencies to particular display selections and select query/display combinations to 'probe' an individual user's degree of 'graphical literacy'. The empirical data is used to instantiate values in the conditional probability tables (CPTs) at each node of the model. The network will then dynamically adjust the CPT values and evolve individualised models for each of its users as they interact with the system. The student model will drive ERST's educational interventions (by hinting or advising) or by 'hiding' inappropriate display forms. The aim is for ERST to be able to generate ER-to-task matching situations that will function as 'probes' of an individual student's knowledge. ERST will be able to interrupt if too much time is spent on selecting a representation (after learning individual's selection display selection latency patterns). It will acquire a basis for recommending the most appropriate display(s) and for varying the range of 'permitted' displays as a function of each task's ER-specificity. If a user manifests a particularly high error rate for particular task/ER combinations, then ERST will be able to offer clarification of, e.g. the functionality of that particular ER. At early stages of learner-system interaction, ERST's adaptiveness will be limited to attempts to offer only display choice options that it believes lie within the learner's 'representational repertoire'. After more extensive learner-system interactions the student model will be more established. At that point ERST may be able to make firmer recommendations to its user and may choose to directly tutor ER-to-task matching skills in the case of ERs for which the student's knowledge appears to be weak.

References

[1] P C-H. Cheng: Functional roles for the cognitive analysis of diagrams in problem solving. In G.W. Cottrell (Eds.): *Proceedings of the 18th Annual Conference of the Cognitive Science Society*, Mahweh, NJ: Lawrence Erlbaum Associates (1996), 207–212.

[2] C. Conati, A. Gertner and K. VanLehn: Using Bayesian networks to manage uncertainly in student modeling. *User Modeling & User-Adapted Interaction*, **12(4)**, (2002) 371–417.

[3] R. Cox: Representation construction, externalised cognition and individual differences. *Learning and Instruction*, **9** (1999), 343–363.

[4] R. Cox and P. Brna: Supporting the use of external representations in problem solving: The need for flexible learning environments. *Journal of Artificial Intelligence in Education*, **6(2)** (1995), 239–302.

[5] R. Cox, P. Romero, B. du Boulay, and R. Lutz: A cognitive processing perspective on student programmers' 'graphicacy'. In A. Blackwell, K. Marriott & A. Shimojima (Eds.): *Diagrammatic Representation & Inference. Lecture Notes in Artificial Intelligence*, Berlin, Springer, (2004).

[6] B. Grawemeyer and R. Cox: A Bayesian approach to modelling user's information display preferences. To be published In: *UM 2005 User Modeling: The Proceeding of the Tenth International Conference*, Berlin, Springer (2005).

[7] L.R. Novick,, S.M. Hurley and M. Francis: Evidence for abstract, schematic knowledge of three spatial diagram representations. *Memory & Cognition*, **27(2)**, (1999), 288–308.

[8] J. Pearl: *Probabilistic reasoning in intelligent systems: Networks of Plausible Inference*. Morgan Kaufmann (1988).

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Towards Data-Driven Design of a Peer Collaborative Agent

Gahgene GWEON, Carolyn ROSÉ, Regan CAREY, Zachary ZAISS
Carnegie Mellon University,
5000 Forbes Avenue, Pittsburgh, PA 15213

Abstract. The research literature investigating the construction of tutorial dialogue and learning companion environments present parallel experiences in attempting to emulate what has been observed to be effective in human-human scenarios. We argue that what is needed as a next step is a careful investigation using controlled experimentation to construct a causal model of how specific features of an agent's behavior influence an individual student's behavior and learning. We outline our research agenda and results from a recent study illustrating our methodology.

1. Motivation

A key aspect of our research is to investigate previous claims about best practices in learning companion design that have not been subjected to rigorous evaluation. We do this using a particular experimental design methodology, which provides a highly controlled way to examine mechanisms by which one peer learner's behavior influences a partner learner's behavior and learning. Specifically, we make use of confederate peer learners who are experimenters acting as peer learners but behaving in a highly scripted way. While this approach lacks the high degree of external validity found in more naturalistic observations of collaborative learning interactions, it provides complementary insights not possible within that framework while allowing for a sophisticated level of agent behavior. We argue that the type of insights provided by this type of design are essential for discovering which combination of technological features will ultimately yield the most desirable response from students. By using a controlled experimental approach, we can get specific information about which aspects of the rich interactions are important for achieving the target effect. The study reported here builds on the work of Hietala & Niemirepo (1998) and Aimeur, Frasson, & Lalond (2001) contrasting high and low performing peer agents. Specifically we address the following questions in the study reported in this paper: Under what circumstances do the errors that arise during collaborative problem solving interactions have a harmful (or helpful) effect on student learning?

Experimental Procedure: The experimental procedure consisted of 5 phases, consisting of three test phases alternating with two instructional phases. The experimental manipulation took place during phase 4. During the pre-instructional testing phase (phase 1), students filled out a consent form, took a pretest to assess their prior domain specific knowledge (for 15 minutes). During the first instructional phase (phase 2), which was a human tutoring phase lasting 45 minutes, students received tutoring on the general concept of differentiation as well as 7 specific rules of differentiation from a human tutor. The tutor was blind to the student's condition and adhered to a rigid schedule for covering all of the content in a consistent way between students. During the mid-instructional testing phase (phase 3), students took a short middle test to assess their learning during phase 2 (for 10 minutes). The second instructional phase (phase 4), was a problem solving phase where students worked through as many of 12 multi-step derivation problems as possible during the allotted 35 minutes. Finally, in the post-instructional phase (phase 5), students took the post-test (for 15 minutes) and filled out a questionnaire.

Assessments: The assessments used to measure learning of derivations consisted primarily of 2 extensive tests (Test A and Test B) that were used for the pre-test (in Phase 1) and the post-test (in Phase 5). These tests each consisted of 7 algebraic manipulation problems, 7 simple calculus problems to test knowledge of each specific differentiation rule, and 6 complex calculus problems requiring both multiple rule applications and algebra. We counterbalanced the order of the tests. In Phase 3, students took a middle test with 7 simple calculus problems, analogous to the second section of tests A and B, and three complex calculus problems requiring multiple rule applications.

Experimental Setup: All on-line problem solving was done using a structured problem solving interface designed for solving differentiation problems. Students first select a rule from a menu. Based on their selection, some explanation about the rule and slots to fill in were presented to the student. In some cases, additional menus were presented, allowing for embedded rule applications. No feedback was provided by the system based on the students' selections from the menu or entries in the text input boxes during the problem solving process. When the student or pairs of students were satisfied with their solution, they submitted it. If it was incorrect, they were then shown their incorrect derivation next to the correct one as a worked example including both the derivation and some explanation.

Experimental manipulation: The experimental manipulation consisted of 4 conditions resulting from a 2X2 full factorial design with two factors describing characteristics of a scripted confederate peer problem solver, namely Lazy(LA)/Engaged(EN), referring to the frequency of the confederate problem solver's contributions to the problem solving process and High(HI)/Low(LO) referring to the accuracy of the confederate peer learner's contributions. During this phase of the experiment, one member of our team acted as a confederate student and another kept track of score, timing, and distribution of labor. The confederate student acted according to the following rules:

- LA/EN: In the Lazy condition, the confederate student contributed to solving the problem either by offering part of the solution in the chat window or by performing an action in the problem solving interface every 45 seconds. In the Engaged, condition, the confederate peer learner contributed every 8 seconds.
- HI/LO: In the High performing condition, the confederate student provided only correct contributions. In the Low performing condition, the confederate student provided incorrect contributions 2/3 of the time.

Subjects: 36 Carnegie Mellon students and staff participated in the study, randomly assigned to conditions: 58% male and 42% female, equally distributed between conditions.

2. Results

Overall, we found a significant interaction effect using an ANCOVA with Post-test scores as the dependent variable, LA/EN and HI/LO as factors, and Pre-test and Middle-test scores as covariates $F(1,30) = 7.47$, $p < .05$, MSE= 7.41. In a post-hoc analysis using a Bonferroni test, the students in the Engaged High performing condition achieved significantly higher post-test scores than the students in the Engaged Low performing condition, $p < .05$. There was a marginal trend in favor of Lazy Low in comparison with Engaged Low $p < .1$. Lazy

High was indistinguishable from the other conditions. Overall we did not find evidence that the errors contributed by the fake peer learner were harmful except in the case of Engaged peer learners.

Because the difference between Lazy Low and Engaged Low was marginal, we wanted to investigate further whether this effect was real or by chance. The strongest predictor of student learning was the number of correct problems the pairs managed to submit during the problem solving phase (CorrectProb). We computed this with a linear regression between CorrectProb and Post-test score with effect of Pre-test score factored out. R-squared=.70, p<.001, N=36. There was a main effect of the HI/LO factor on the number of correct solutions contributed, with the effect of Pre-test and Mid-test scores used as covariates, F(1,30) = 49.1, p < .001, MSE=.93, effect size = 2.4 standard deviations. Since there was a strong correlation between Mid-test score and correct problems contributed, we replaced mid-test with correct problems submitted as a covariate in the original ANCOVA with LA/EN and HI/LO as factors. We used Pretest score and Correct Problems submitted as covariates. While Pretest and CorrectProb submitted together explain about 71% of the variation in post test scores across our student population, we still found a significant crossover interaction effect explaining an additional 4% of the variance that provided some weak evidence that the errors contributed by the fake peer learners sometimes had a positive effect on student learning above and beyond the effect of correct solutions submitted. F(1,30) = 4.96, p<.05, MSE=10.68. On the continuum between High and Low performing peer learners, students in the Lazy condition learned more when the peer learner contributed more errors, whereas the trend was the opposite with Engaged peer learners.

3. Conclusions and Current Directions

The results of this investigation contribute insights towards a detailed causal model of how environmental factors influence student behavior and learning. An understanding of where errors can be used strategically to stimulate cognitive conflict and student learning may enhance the effectiveness of existing well-established approaches to scaffolding in intelligent tutoring systems. Nevertheless, this is an issue that requires more investigation. Because the majority of the observed learning in this study is explained by correct problem solving, these results do not argue that errors play a large role in student learning relative to correct examples. The weakness of this effect might be explained by a paucity of what is referred to as “high level” explanation and help seeking behaviors found in our corpus of collaborative problem solving interactions [3]. Webb et al. found, for example, that high ability students only benefited from their interactions with lower ability peers when their group engaged in high level explanation and help-seeking behaviors. We plan to do more investigations along these lines and to use the results to eventually inform the design of a new peer collaborative agent.

4. Acknowledgements

This work was funded by NSF SGER REC-0411483 and NSF/IERI REC-043779.

References

[1] Aimeur, E., Frasson, C., Lalonde, M. (2001). The Role of Conflicts in the Learning Process, *SIGCUE OUTLOOK* 27(2).

[2] Hietala, P. & Niemirepo, T. (1998). The Competence of Learning Companion Agents. International *Journal of Artificial Intelligence in Education*, 9, pp178-192.

[3] Webb, N., Nemer, K., & Zuniga, S. (2002). Short Circuits or Superconductors? Effects of Group Composition on High-Achieving Students’ Science Assessment Performance, *American Educational Research Journal*, 39, 4, pp 943-989.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Discovery of Patterns in Learner Actions

Andreas Harrer, Michael Vetter, Stefan Thür, Jens Brauckmann,
[a] *Institut für Informatik und interaktive Systeme, Universität Duisburg-Essen*

Abstract. This paper describes an approach for analysis of computer-supported learning processes utilizing logfiles of learners' actions. We provide help to researchers and teachers in finding insightful patterns of the learning process in two ways: patterns can be specified explicitly to be searched in logfiles or automatic extraction of patterns complying to configurable parameters can be initiated to find the most typical sequences within the logfile. Both support features have been implemented in a stand-alone tool that accepts generic logfiles usable with a potentially wide variety of different learning systems.

In recent years the importance of additional functionality for active support of all people involved in CSCL learning, i.e. students, teachers, and researchers, has grown. Learning Protocols are readily available in most CSCL-systems simply by logging the activities that take place in the collaborative environment. Logfiles are used in various ways related to the goals of the AIED area. They have been used for manual inspection and interpretation [1], for exploration of mis-use of tutor support [2] and even for construction of tutors from real data [3]. These logfiles are usually at a very low abstraction level and of little use for teachers and learners. Given that we want to help users of CSCL systems to better understand the learning processes by providing a tool that enables the user to analyse logfiles with different kind of CSCL systems.

1. Extraction of Patterns from Logfiles

Typical sequences of user actions can indicate phases of collaboration or situations of the learning process, such as turn taking between learners or communication breakdowns. A support to find and visualize these sequences helps the teacher in evaluating the process, the researcher in understanding the learning, and the student when reflecting her own activities. For that sequences have to be searched for matching a specific *pattern*, so we will use the term pattern for a typical sequence of user actions. In logfiles of collaborative learning sessions usually actions on coordination level, like chat messages, are interspersed with domain-related actions, such as creating a UML-Class. Thus the issue how to find patterns that are not strictly cohesive in the logfile has to be addressed in a tool supporting the user in logfile analysis. Another topic to consider when designing a tool for the logfile analysis is that often the researcher has some hypothesis but cannot clearly specifiy how this hypothesis manifests itself on the low level of a computer-generated logfile. For this it is desirable to automatically compute frequently occuring patterns that have not been specified as a query beforehand.

1.1. Extraction of Specified Patterns

Specified patterns can be found using standard algorithms or query engines in case that the patterns are cohesive in the logfile. Otherwise the complexity of finding patterns is increased because incomplete instances of patterns can be continued in later parts of the logfiles. Because of this we offer to filter out activities that are not relevant if interspersed in a specified pattern by choosing for each type of action in the logfile (according to action target and the type of action) if it shall be considered. The same can be done for specific users. Specification of patterns can be done by the user in three different ways:
Rule specification by action types and rule composition: Our tool shows in a list all action types (combination of object types and respective actions) that occur in the logfile. The user chooses action types from the list in the desired sequence. Dependencies between the elements of a rule, such as "search for an UML-Class object that is created and later renamed", are specified by identical variables in that slot, here the Object slot. Rules can be used as components of new rules to compose more complex rules.
Specification by Example: The user also has the option to define a rule in a more informal way by selecting directly actions from the logfile as the pattern searched for. The tool abstracts from the concrete values of the different slots of actions, but keeps the dependencies using the same variables for identical values.
Direct editing of rules in the query language: For the expert user there is also the option to directly edit and modify the rules (either created by rule specification or by example) at the level of the query language, in this tool Prolog. This allows the user to add more constraints to the rule manually, but requires more expertise than the other methods, that hide the implementation level from the user.

Regardless of the method chosen for specification of rules the rules can be compiled into rule sets that can be stored for later re-use. Thus a researcher can load a previously defined rule set, which compiles his standard rules for logfile analysis, then choose the rules to be used for the pattern search.

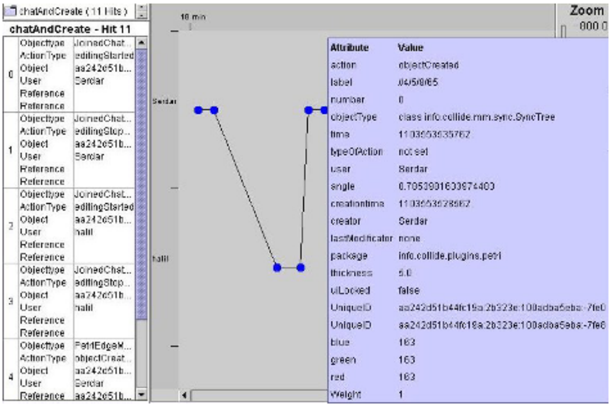
1.2. Finding Unspecified Patterns

The task of providing typical patterns that have not been specified before, demands a different algorithmic solution. For this we chose the "sawtooth" algorithm [4] that scans a string for the longest re-occuring substring starting from each position. An un-supervised run will produce a lot of potential patterns of unspecified length and frequency; additionally this algorithm considers only directly succeeding patterns. To produce better results we provide options for a user-guided pattern mining, i.e. he defines general properties the patterns searched for have to comply to. These options are:
Filtering of Objects, Users, and Action types: The user can review all the actions that have been conducted with this specific object, by one user, or of the same type, and choose if this should be considered for the pattern discovery run.
Object context: If the CSCL system producing the logfile allows relations between objects (e.g. Belvedere, CoLab, CoolModes, ModellingSpace), it is possible to consider also the semantic context of objects expressed by the related objects. By configuring the maximally considered relational distance from the object the size of the context can be varied; e.g. choosing 2 for the distance means, that all objects connected via one or two relations to the object will also be used.

Pattern length and Frequency: Only patterns of a minimum length and surpassing a specified number of occurrences will be produced by the search run.

By combining several of these options the user guides the pattern discovery process in the direction of the patterns he is interested in. The pattern discovery produces rules, each characterizing one found pattern schema fulfilling all the option's criteria, which can be used as if they have been specified by the user directly (cf. previous subsection).

This concept has been implemented within a tool and put to practical use with logfiles of student pairs using our modelling environment CoolModes. As proof-of-concept the unspecified discovery produced typical sequences for collaboration, such as "'chat message exchange followed by a construction phase"', in logfiles that we also analysed manually. These results were encouraging (see figure 1), though the rules for unspecified patterns must be fine-tuned by user's parameterization to find the characteristic patterns.



chatAndCreate - Hit 11

0	Objecttype	JoinedChat...
	ActionType	editingStarted
	Object	aa242d51b...
	User	Serdar
	Reference	
	Reference	
1	Objecttype	JoinedChat...
	ActionType	editingStop...
	Object	aa242d51b...
	User	Serdar
	Reference	
	Reference	
2	Objecttype	JoinedChat...
	ActionType	editingStarted
	Object	aa242d51b...
	User	halil
	Reference	
	Reference	
3	Objecttype	JoinedChat...
	ActionType	editingStop...
	Object	aa242d51b...
	User	halil
	Reference	
	Reference	
4	Objecttype	PetriEdgeM...
	ActionType	objectCreat...
	Object	aa242d51b...
	User	Serdar
	Reference	aa242d51b...

Attribute	Value
action	objectCreated
label	.04/5/8/65
number	0
objectType	class info.collide.mm.sync.SyncTree
time	1103553935762
typeOfAction	not set
user	Serdar
angle	0.7053901603974403
creationtime	1103553978567
creator	Serdar
lastModificator	none
package	info.collide.plugins.petri
thickness	5.0
uiLocked	false
UniqueID	aa242d51b44fc19a:2b323e:100adba5eba:-7fe0
UniqueID	aa242d51b44fc19a:2b323e:100adba5eba:-7fe6
blue	163
green	163
red	163
Weight	1

Figure 1. Visualization: Timeline (middle) and pattern instance (left) for "'chat exchange and construction"'. The pattern was produced by unspecified discovery and named by the user according to its meaning.

The analysis tool is designed to accept not only these logfiles, but provides a generic logfile format and transformation functionality to this format, given a suitable XSL-T script to map the logfiles of arbitrary learning systems to the generic format. We also plan to use our tool in the future for analysis of lab and classroom experiments, as we did mainly manually in previous studies.

References

[1] De Vries, E., K. Lund, M.J. Baker. Computer-mediated epistemic dialogue: Explanation argumentation as vehicles for understing scientific notions. *The Journal of Learning Sciences*, 11(1):pp. 63–103, 2002.

[2] Baker, R.S., A.T. Corbett, K.R. Koedinger. Detecting student misuse of intelligent tutoring systems. In *Proc. of ITS2004*. J.C. Lester, R.M. Vicari F. Paraguaçu, 2004.

[3] McLaren, B.M., K.R. Koedinger, M. Schneider, A. Harrer, L. Bollen. Bootstrapping novice data: Semi-automated tutor authoring using student log files. In *Proc. of Workshop on Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes, ITS2004*, 2004.

[4] Hoppe, U., R. Plötzner. Inductive knowledge acquisition for a unix coach. In *Informatics Psychology Workshop*, pages 313–335, 1989.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

When do Students Interrupt Help? Effects of Time, Help Type, and Individual Differences

Cecily HEINER, Joseph BECK, and Jack MOSTOW
Carnegie Mellon University
Project LISTEN, Newell Simon Hall, 5000 Forbes Avenue Pittsburgh, PA 15213

Abstract. When do students interrupt help to request different help? To study this question, we embedded a within-subject experiment in the 2003-2004 version of Project LISTEN's Reading Tutor. We analyze 168,983 trials of this experiment, randomized by help type, and report patterns in when students choose to interrupt help. Using the amount of prior help, we fit an exponential curve to predict interruption rate with an r^2 of 0.97 on aggregate data and an r^2 of 0.22 on individual data. To improve the model fit for individual data, we adjust our model to account for different types of help and individual differences. Finally, we report small but significant correlations between a student parameter in our model and external measures of motivation and academic performance.

1. Introduction

One of the key benefits of an intelligent tutoring system is on-demand help. However, students do not always appreciate the help that a tutoring system gives, and students may interrupt some forms of help in order to receive more desirable help. In this paper, we study when students interrupt help and we report results which may assist in designing better research and better help systems.

We describe work that relates interruption in a user-system dialogue, help seeking, and user modeling. Earlier research has reported on a system that interrupts when a student deviates from correct behavior, hesitates, or gets stuck[1], but interrupting a user has a cost [2]. When students interrupt the system, they are often seeking help, a behavior that can be predicted with 83% accuracy [3]. However, researchers have found that students often do not know when they need help[4], and they have identified help abuse as a problem that accounts for a third or more of help seeking bugs[5]. One particular example of help abuse is described as “gaming the system”[6], a pattern in which some students “click through” help until they receive the answer with negative consequences to their learning. Extending the idea that help-seeking behaviors relate to the attributes of a specific student, researchers have inferred student variables based partially on help-seeking and help-usage behavior[7]. However, a more integrated approach to understanding help-seeking behavior and user-modeling could improve both our student models and our assessments of what they know[8]. In this paper, we present an experiment and series of analyses that examines interruption and user-system dialogue, help seeking behavior, and user modeling together.

In this paper, we present an experiment, first explaining the design behind the experiment and then describing the data set. Next we analyze the results of that experiment and use them to fit a series of models. Then we interpret the results and correlate them with other data. Finally, we conclude and suggest future work.

2. Experimental Design

Our data come from the 2003-2004 version of the Project LISTEN Reading Tutor, which presented text and used automatic speech recognition to listen to children read aloud[9]. When a student encountered a difficult word, the student could click on that word for help. Alternatively, the Reading Tutor may have detected that a student was struggling with a word and taken the initiative to give spoken or graphical help. Regardless of whether the help was student initiated or tutor initiated, the student could interrupt that help and receive more help by clicking on the word again while the Reading Tutor was speaking. Sometimes, students clicked two or more times, interrupting the previous help with each click and triggering a new instance of help. Alternately, a student may have clicked on a word for help more than once, but waited until the previous help had completed before clicking again. Each time the Reading Tutor gave help, it chose randomly from a variety of applicable and efficacious help types without regard to previous help that it had given. The Reading Tutor primarily gave the nine types of word help listed below[10].

- ***SayWord*** plays a recording of the word. e.g. "cat"
- ***WordinContext*** plays a recording of the word extracted from the sentence.e.g. "…cat…"
- ***Autophonics*** pronounces a selected grapheme in the word. e.g. "c here makes the sound /k/"
- ***SoundOut*** plays video clips of a child's mouth saying the phonemes of the word. e.g. "/k/…/ae/…/t/"
- ***Recue*** reads words in the sentence leading up to, but not including, the word. e.g. "I have a dog and a"
- ***OnsetRime*** says the first phoneme, pauses, and says the rest of the phonemes. e.g. "/k/…/ae/ /t/"
- ***StartsLike*** says "starts like (word with the same beginning)." e.g. "starts like cats"
- ***RhymesWith*** says "Rhymes with (rhyming word)." e.g. "rhymes with mat"
- ***Syllabify*** says the syllables of the word separated by short pauses. e.g. "cat"

The Reading Tutor randomized the choice to provide a variety of help [1] and to embed an experiment to compare the effectiveness of different types of help[11].

This embedded experiment examines when students interrupt help. Each randomized trial starts when the student or tutor initiates help. The randomized variable is the selection of help type selection. Another analysis[10] considered students' subsequent performance when reading the word as the outcome variable. In this experiment, the outcome variable is whether or not students interrupt help. The experiment is diagrammed in Figure 1.

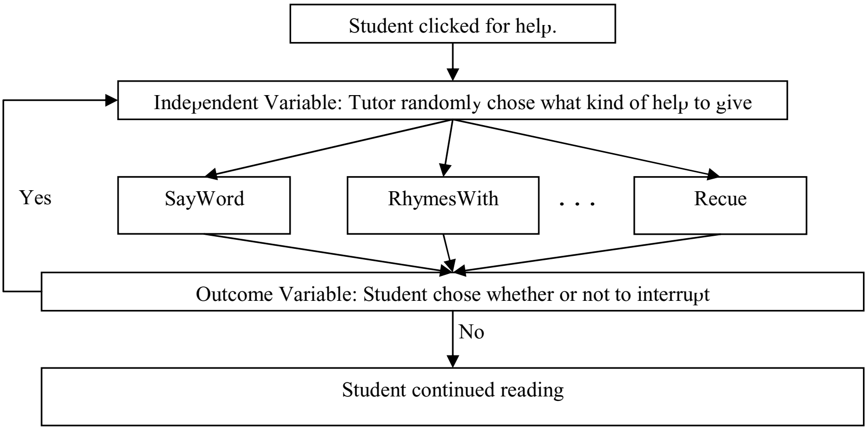

Student clicked for help.
Independent Variable: Tutor randomly chose what kind of help to give
Yes
SayWord
RhymesWith
. . .
Recue
Outcome Variable: Student chose whether or not to interrupt
No
Student continued reading

Figure 1: Experiment Diagram

3. Modeling Approach

We considered prior exposures to help as an important independent variable in our experiments. In this paper, we define prior exposures to help as the number of times a particular user has received help of a particular type before. Because not all students receive the same amount of exposure to the Reading Tutor, we were concerned that maybe some students interrupted more because they had more exposure to the Reading Tutor; in other words we thought maybe we were seeing an effect of attrition. To insure that our trend could not be explained by attrition, we included the first hundred trials for a given student and help type, excluding students and help types with fewer. We chose 100 because Figure 2 showed that it was large enough to reveal an asymptote, but not so large that it would eliminate data unnecessarily or create a bias favoring certain models. This step left data from 368 K-4 students with a variety of reading abilities. We did not distinguish between tutor-initiated (15% of the data) and student-initiated (85% of the data) help or exclude students who had not received all nine help types from this data set.

3.1 Fixed Parameter Model

Plots of the data in Figure 2 show the exponential relationship in Equation 1. To estimate the values of the parameter *a*, *b*, and *c* of the equation accurately, we used SPSS[12], software for statistical analysis, to do a non-linear regression analysis. We used a Java applet [13] to initial parameter estimates for a curve with a shape similar to our data. Using the initial parameters, SPSS found that a= -0.27, b= -0.08, and c=0.45 in this model. After considering other models including a power curve and a logarithmic model, we selected the exponential curve because other researchers have suggested that it is a better fit for individual data [14], and it fit our data best.

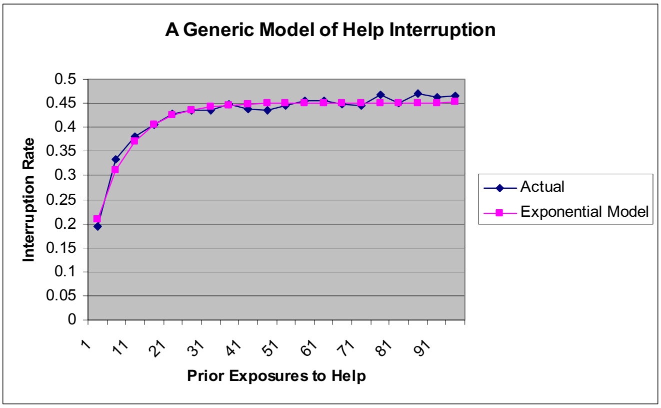


Figure 2: **Simple Model**

$$P(i) = a * (e^{b * prior\_exposures}) + c$$

Equation 1: Model Form

As shown in Figure 2, the actual data and the exponential model are correlated. The r^2 of .97 shows that the aggregate data fit the model well. We speculated that this equation may be a learning curve for recognizing undesirable help. We also speculated that the parameters in this equation are related to properties of a specific help type or the student. We caution that this model is fit using aggregate data; a model fit to an individual student's data with a binary outcome variable will have a much lower r^2, 0.22 for this dataset. For both the aggregate and the individual data, the asymptote represents the interruption rate after students have developed habits that alter how they use the system. Because interruption rates vary by help type, we believed we could add conceptual value to the model and improve its fit for individual data by adding parameters to account for help type and individual differences.

3.2 Fitting a Model with Help Type Parameters

To build a model that was more closely related to help type, we adjusted parameters *a*, *b*, and *c* by refitting them for each help type, instead of aggregating all the help types together. To estimate the parameters of the equation for each help type, we did another SPSS non-linear regression. Using generic values as initial estimates of the parameters, SPSS found the values in Table 1.

Table 1: Help Types

Help Type	*a*	*b*	*c*	Interruption Rate	Duration
SayWord	-.11	-.06	.14	.12	.69 s
WordInContext	-.10	-.21	.19	.18	1.11 s
OnsetRime	-.29	-.07	.39	.35	1.19 s
SoundOut	-.41	-.06	.55	.47	2.91 s
Syllabify	-.39	-.15	.54	.52	2.00 s
RhymesWith	-.33	-.08	.59	.55	2.36 s
StartsLike	-.38	-.11	.61	.58	2.60 s
Autophonics	-.49	-.09	.75	.69	2.43 s
Recue	-.31	-.07	.79	.75	3. 89s

Now we had nine values for each parameter and we could look for patterns. When we correlated our *c* parameter with the interruption rate for a help type we found an r^2 of .99. Independent of the model, another clear pattern relates the interruption rate for a particular help type and duration with an r^2 of .79. Thus the *c* parameter models the average interruption rate for a particular type of help which is related to the duration of help. This makes sense since students have more time to move their mouse and click to interrupt when help has a longer duration. We did not find clear patterns for *a* and *b*, but we hypothesize that that *a* and *b* may related to how many exposures a student needs to learn to recognize a specific help type and interrupt it because these parameters are lower for the help types that give the answer; we do not currently have the data and analysis to confirm this idea.

3.3 Fitting a Model with a Student Parameter

To better account for individual differences, we added one more parameter to our equation. We held the help type parameters of the equation at their values from the previous regression and estimated a new student parameter *s,* applying SPSS non-linear regression to Equation 2. Conceptually, this student parameter, *s*, alters the asymptote of the graph and is related to a student's interruption rate, a value that should be between zero and one. To insure that values for *s* would be consistent with this idea, we altered the form of the model slightly, setting the initial value for the student parameter *s* at 1, and imposing the limits that *s* must be less than or equal to 1 and greater than -.5. Within this range [-.5, 1], SPSS fit a single student parameter for each student.

$$P(i) = a * (e^{b*opportunities}) + c + (\mathrm{s} * (1 - c))$$

Equation 2: Student Parameter Model

3.4 Evaluating the Relative Value of the Various Models

Table 2 compares the various models and two additional baseline models, using mean squared errors and r^2. The overall interruption rate model simply predicts that 43% of all help will be interrupted, since this is the average interruption rate when all of the data is aggregated together. The mean interruption by help type model predicts help interruption based on the interruption rate for a given help type. We included both of these baselines to measure how much variance the help type accounts for on its own. Table 2 shows that the biggest reductions in mean square error and improvements in r^2 come from applying a generic model that takes time into consideration by accounting for the amount of previous help. Fitting the model based on help type improved the model a little, but not much. Adding a student parameter improved the model moderately.

Table 2: Models and Mean Square Errors

Model Name	Mean Square Error	r^2
Overall Interruption Rate	.24	-
Mean Interruption by Help Type	.24	0.01
Generic Model with Prior Help	.19	0.22
Help Type Parameters	.19	0.24
Student / Help Type Parameters	.17	0.30

4. Correlating the Student Parameter against External Measures

The student parameter, *s* in the final model is a variable that may relate to other measures of a student, including process variables and test scores. We considered the following process variables: help request rate, help interruption rate, disengagement (measured as the percentage of questions that students answer hastily[15]), and percentage of time picking stories. We were surprised that we did not find correlations with other affective variables such as disengagement or help request rate.

For test scores, we considered pre- and post-test scores and gains for the Elementary Reading Attitude Survey (ERAS) [16] and a fluency test. ERAS is a twenty item instrument with ten items each for recreational and academic reading attitudes. The fluency test consists of a timed, levelled reading passage which students read for trained fluency testers. Small, significant negative correlations exist between ERAS academic and motivational test scores. So, *s* relates to attitudes towards academic and recretaional reading. Additionally, small but insignificant correlation exists between fluency pre-test and the student parameter *s*. So, *s* may also be related to fluency. Table 3 displays the meaningful correlations.

Table 3: Student Parameter Correlations

Test Name	Pearson Correlation	Signficance
Fluency Pre-Test	-.155	.072
ERAS Recreational Pre-Test	-.267	.002
ERAS Academic Pre-Test	-.283	.001

In order to determine the relationship between *s* and gender, we ran an independent T-test and found the mean *s* value for girls is -0.057 and the mean *s* value for boys is 0.037 with a p-value of <0.001. This means that girls are less likely to interrupt than boys, the difference is significant, and the *s* parameter is related to gender.

Future Work and Conclusion

This paper is the first to study when students interrupt spoken help and to propose a predictive model of this behavior. An exponential model characterizes the temporal aspect of this behavior and shows that the number of previous exposures to a particular type of help is an important predictor of whether or not a student will interrupt help. We report values for three parameters that characterize help type and show that one of them correlates highly with the interruption rate for a given help type. Additionally, girls are less likely to interrupt than boys. Interruption rates are somewhat negatively correlated with pre-test scores, so less motivated poor readers interrupt more. We compare successively refined models for predicting help interruption rate; the biggest improvement in model fit comes from accounting for temporal factors. The exponential model could be a learning curve for recognizing help that students find undesirable.

This paper has illuminated how students use help. We have suggested that there is an initial window of adaptation when students are learning to use an intelligent tutoring system. After this window, students interrupt each help type at an approximately constant rate. In our model, the interruption rate approaches the asymptote when the student has had approximately thirty prior encounters with a particular kind of help. Thirty prior encounters of help roughly corresponds to an average of three hours of system usage spread across eighteen sessions or six weeks of calendar time. These patterns suggest a need for long-term studies to understand how students use intelligent tutoring systems after they have adapted to them. We have also proposed that initial data should be considered separately due to startup effects.

This paper is one step towards the long-term goal of being able to quantify affective factors and link them to learning gains. We still do not know very much about why students interrupt help. Are they bored, tired, lazy, impatient, or rude? What are students looking for when they interrupt help? The answers to these questions might suggest how we can encourage students to tolerate long, laborious, but educational help.

Acknowledgements

This work was supported by the National Science Foundation, under ITR/IERI Grant No. REC-0326153. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation or the official policies, either expressed or implied, of the sponsors or of the United States Government.

Thanks to Ryan Baker, Hao Cen, and Jonathan Brown for comments on drafts of this paper. Additional thanks to students, teachers, and administrators in schools where the Reading Tutor is deployed.

References (website: www.cs.cmu.edu/~listen/)

1. Mostow, J. and G. Aist, *Giving help and praise in a reading tutor with imperfect listening -- because automated speech recognition means never being able to say you're certain.* CALICO Journal, 1999. **16**(3): p. 407-424.
2. Horvitz, E. and J. Apacible. *Learning and Reasoning about Interruption*. in *ACM International Conference on Multimodal Interfaces*. 2003. Vancouver, Canada.
3. Beck, J.E., P. Jia, J. Sison, and J. Mostow. *Predicting student help-request behavior in an intelligent tutor for reading*. in *Proceedings of the 9th International Conference on User Modeling*. 2003. Johnstown, PA.
4. Aleven, V. and K.R. Koedinger. *Limitations of Student Control: Do Student Know when they need help?* in *Proceedings of the 5th International Conference on Intelligent Tutoring Systems*. 2000. Montreal: Berlin: Springer Verlag.
5. Aleven, V., B. McLaren, I. Roll, and K.R. Koedinger. *Toward tutoring help seeking Applying cognitive modeling to meta-cognitive skills*. in *Intelligent Tutoring Systems*. 2004. Maceio, Alagoas, Brazil: Springer.
6. Baker, R.S., A.T. Corbett, K.R. Koedinger, and A.Z. Wagner, *Off-Task Behavior in the Cognitive Tutor Classroom: When Students "Game The System"*. Proceedings of ACM CHI, 2004: p. 383-390.
7. Arroyo, I., T. Murray, and B.P. Woolf. *Inferring observable learning variables from students' help seeking behavior*. in *Intelligent Tutoring Systems. Workshop on Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes*. 2004. Maceio, Brazil.
8. Wood, H. and D. Wood, *Help seeking, learning, and contingent tutoring.* Computers & Education, 1999. **33**(2-3): p. 153-169.
9. Mostow, J. and G. Aist, *Evaluating tutors that listen: An overview of Project LISTEN*, in *Smart Machines in Education*, K. Forbus and P. Feltovich, Editors. 2001, MIT/AAAI Press: Menlo Park, CA. p. 169-234.
10. Heiner, C., J.E. Beck, and J. Mostow. *Improving the Help Selection Policy in a Reading Tutor that Listens*. in *Proceedings of the InSTIL/ICALL Symposium on NLP and Speech Technologies in Advanced Language Learning Systems*. 2004. Venice, Italy.

11. Heiner, C., J. Beck, and J. Mostow. *Lessons on using ITS data to answer educational research questions*. in *Proceedings of the ITS2004 Workshop on Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes*. 2004. Maceio, Brazil.

12. SPSS, *SPSS for Windows*. 2000, SPSS Inc.: Chicago, IL.

13. Dendane, A., *Experiment and Explore Mathematics: Tutorials and Problems*. 2005.

14. Heathcote, A., S. Brown, and D.J.K. Mewhort, *The Power Law Repealed:The Case for an Exponential Law of Practice.* Psychonomics Bulletin Review, 2000: p. 185-207.

15. Beck, J.E. *Using response times to model student disengagement*. in *Proceedings of the ITS2004 Workshop on Social and Emotional Intelligence in Learning Environments.* 2004. Maceio, Brazil.

16. Kush, J.C. and M.W. Watkins, *Long-term stability of children's attitudes toward reading.* The Journal of Educational Research, 1996. **89**: p. 315-319.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Fault-Tolerant Interpretation of Mathematical Formulas in Context

Helmut Horacek [a] Magdalena Wolska [b]
[a] *Fachrichtung Informatik*
[b] *Fachrichtung Computerlinguistik*
Universität des Saalandes, Postfach 15 11 50, D-66041 Saarbrücken, Germany
{horacek@cs.uni-sb.de,magda@coli.uni-sb.de}

Abstract. We present a fault-tolerant formula interpreter that aims at finding plausibly intended, formally correct specifications from student statements containing formal inaccuracies. Its methods comprise local changes based on error categories, fault-tolerant structure building, and testing contextually-motivated alternations.

1. Categories of Errors and Associated Correction Attempts

In tutorial systems, effective elaboration of the problem-solving target is frequently hindered by formal sloppiness and low-level errors made by the student. With our fault-tolerant formula interpreter, we attempt to reveal probably intended specifications. This work is part of the DIALOG project [1] [1], which aims at teaching proofs of mathematical theorems. For checking validity of possibly ambiguous proof-step interpretations and consistency within the proof context, we use the proof development environment ΩMEGA [2]. Interaction with ΩMEGA is mediated by a *Proof Manager*, whose task is to build and maintain a representation of the constructed proof, as well as a discourse memory of identifiers and operators. To investigate phenomena characterizing written computer-based interaction with an automated tutor, we collected a corpus of tutor-student dialogs in a Wizard-Of-Oz experiment in the domain of naive set theory [4].

In Table 1, we present examples of flawed formula from this corpus. In (1), a segmentation error is shown: not only a space between the operator symbol P and identifier C, but also parentheses are missing. (2) is an example of a typing error, where an operator symbol p has been used in place of an identifier b. In (3), the types of arguments of the main operator are invalid. (4) shows a well-formed formula, but it is not relevant in the context of the task: a stronger assertion about an intersection rather than union of the sets on the right-hand side of the equation was expected. In (5), similarly to (4), a weaker assertion of set inclusion ($\subseteq$) rather than equality is expected. Finally, (6) and (7) are examples of commonly confused relations of *subset* and *membership*.

Finding purposeful changes in a formula that aim at building a corrected and possibly intended version of that formula, is done differently for 1) *logical*, 2) *type*, and 3)

[1] The DIALOG project is part of the Collaborative Research Center on *Resource-Adaptive Cognitive Processes* (SFB 378) at University of the Saarland: http://www.coli.uni-sb.de/sfb378/.

Table 1. Examples of flawed formulas from the corpus

	Example Formula	*Error Category*
(1)	$P((A \cup C) \cap (B \cup C)) = PC \cup (A \cap B)$	3
(2)	$(p \cap a) \in P(a \cap b)$	2
(3)	$(x \in b) \notin A \qquad x \subseteq K(A)$	2
(4)	$P((A \cup C) \cap (B \cup C)) = P(A \cup C) \cup P(B \cup C)$	1
(5)	$P((A \cap B) \cup C) = P(A \cap B) \cup P(C)$	1
(6)	$(A \cap B) \subseteq P(A \cap B)$	1
(7)	if $A \subseteq K(B)$ then $A \notin B$	2

structural errors: 1. the formula analyzer cannot build an analysis tree on the basis of the defined constructors (*error category 3*), or 2. it cannot resolve a type mismatch in an analysis tree built successfully (*error category 2*), or 3. the proof manager evaluates a correctly analyzed formula as a *wrong* statement (*error category 1*). In case of an error, attempts are made to remedy the committed error by applying local and contextually justified modifications to the formula. In order to obtain meaningful changes, we associate a set of replacement rules with each error category, aiming to achieve an improvement of at least one category level. Some rules developed on the basis of errors observed in the corpus, their associated error categories, and examples are illustrated in Table 2.

2. The Formula Modifying Algorithm

The formula modification procedure presented here consists of two parts: a mildly fault-tolerant parser and a formula modification tester. The parser extends the method of parsing mathematical expressions embedded within natural language text [3]. Extended formula analysis consists of three stages: (1) mathematical expressions are identified within word-tokenized text; (2) the identified sequence is verified as to syntactic validity and, in case of a parentheses mismatch, a correction procedure is invoked, thereby implementing replacement rules of category 3, and (3) the expression is parsed. The tagger has access to a list of operation and identifier characters relevant in the given context. Identification of mathematical expressions is based on simple indicators: single character tokens (including parenthesis), multiple-character tokens consisting only of known relevant characters, mathematical symbol unicodes, and new-line characters. Multiple-character candidate tokens are further segmented into operators and identifiers by inserting the missing spaces. Once a candidate string is identified, "chains" of formulas are separated into individual segments. Furthermore, missing parentheses are inserted while observing the following preferences for the resulting expressions: (i) provide parentheses for operators that require bracketed arguments, (ii) avoid redundant parentheses (i.e. double parentheses around the same substring). Syntactically correct candidate sequences are parsed by a tree-building algorithm. The algorithm has access to standard requisite information such as: list of operators and operator precedence. The output of the parser is a set of formula trees with nodes marked as to type compatibility and bracketing where applicable.

The formula modification tester starts with the set of formula trees obtained, incrementally generating formula alternations by applying replacement rules in a best-first fashion and testing their impact on resolving the original error. Successors are generated

Table 2. Replacement rules attempting to remedy errors

Replacement Rules	*Error Categories*	*Examples* (set theory)
dual operators	1	$\cap \Leftrightarrow \cup$, $\subset\Leftrightarrow\supset$, $\subseteq\Leftrightarrow\supseteq$
stronger/weaker operators	1	$\supset\Leftrightarrow\supseteq$, $\subseteq\Leftrightarrow=$, $\supseteq\Leftrightarrow=$
confused operators	1,2	$\subset\Leftrightarrow\in$, $K \Leftrightarrow P$
confused identifiers	1,2	$a \Leftrightarrow b$, $P \Leftrightarrow a$, $P \Leftrightarrow b$
delete character	2	$Pc \Rightarrow P$, $Pc \Rightarrow c$
insert a pair of parentheses or a blank	3	$Pc \Rightarrow P(c)$, $Pc \Rightarrow P\ c$

for the formula considered best by applying replacement rules in the category associated with the error reported for the original formula, unless the error is already resolved. If a maximum number of modified formulas and a time limit are not exceeded, the most promising successor of a formula generated so far is promoted into the new best one, its correctness state is assessed by the proof manager, and successor generation is repeated. Otherwise, an ordered list of modified formulas examined is returned. Preferred orderings among created formulas are established by the error-related category and a similarity-assessing function, the former dominating the latter. The assessment function combines the number of replacement rules applied and the number of structural differences to the most similar formula in the context, which comprises the set of formulas consisting of the goal expression, the previous proof step and possible follow-up steps.

3. Results

For utterance (1), we get two interpretations, depending on whether PC is separated by inserting parentheses (2 alternatives), or flagged as a type error. Replacing PC by any type compatible identifier yields error category 1. The same holds for the parenthesis insertion with narrower scope, $P(C)$, but the other alternative, $P(C \cup (A \cap B))$ yields no error and wins. For utterance (2), only replacing the first occurrence of P flagged as a type clash is subject to being changed. Only replacements by A and B yield no error, B winning over A since it gets a better context agreement count. In utterance (5), changing variables gives lower agreement scores than changing an operator in their dual counterpart, but among all these choices only replacing $=$ by $\supseteq$ yields a correct assertion.

References

[1] C. Benzmüller et al. Tutorial Dialogs on Mathematical Proofs. In *Workshop on Knowledge Representation and Automated Reasoning for E-Learning Systems at the 18th International Joint Conference on Artificial Intelligence*, pages 12–22, Acapulco, Mexico, 2003.

[2] J. Siekmann et al. Proof Development with ΩMEGA. In *Proceedings of the 18th Conference on Automated Deduction (CADE-02)*, pages 144–149, Copenhagen, Denmark, 2002.

[3] M. Wolska and I. Kruijff-Korbayová. Analysis of Mixed Natural and Symbolic Language Input in Mathematical Dialogs In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pp. 24–32, Barcelona, Spain, 2004.

[4] M. Wolska et al. An Annotated Corpus of Tutorial Dialogs on Mathematical Theorem Proving. In *Proceedings of 4th International Conference on Language Resources and Evaluation (LREC-04)*, pp. 1007–1010, Lisbon, Portugal, 2004.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Help in Modelling with Visual Languages

Kai Herrmann, Ulrich Hoppe, and Markus Kuhn
University of Duisburg-Essen, Germany.
{herrmann, hoppe, kuhn}@collide.info

Abstract. In this paper we describe how a constraint checking system can provide advanced help for a visual language modelling process, not only taking into account single elements, but also the context of the element for which help is needed.

1. Introduction: Help for Visual Languages

The Cool Modes modelling software [1] provides a multi-workspace environment in which users can construct models using visual languages, consisting of domain related elements with specific semantics (stochastic experiments, petri nets). Visual languages differ from textual representations, e.g. [2], in the fact that not only the *values*, but also the very *existence* and the *location* of these objects are of importance. Users construct models composed of graph structures, which, as a whole, form expressions in visual languages. Providing help for such languages is a crucial challenge. A help system for a visual language must take into account the context of an object: while a text field in a dialog always has the same function, an element in a visual language can have different

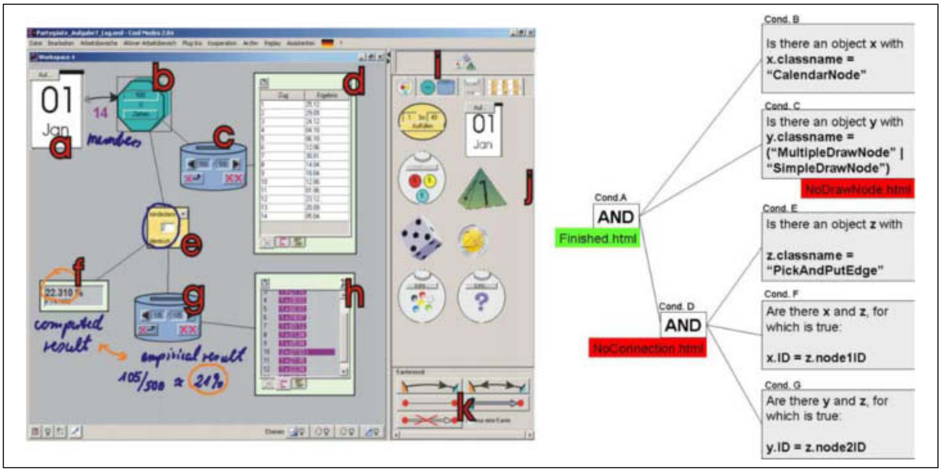

01
Jan
a
b
14
numbers
c
d
e
f
22.310 %
computed result
g
empirical result
105/500 ≈ 21%
h
i
j
k
Cond. B
Is there an object x with x.classname = "CalendarNode"
Cond. C
Is there an object y with y.classname = ("MultipleDrawNode" | "SimpleDrawNode")
NoDrawNode.html
Cond.A
AND
Finished.html
Cond. E
Is there an object z with
z.classname = "PickAndPutEdge"
Cond. D
AND
NoConnection.html
Cond. F
Are there x and z, for which is true:
x.ID = z.node1ID
Cond. G
Are there y and z, for which is true:
y.ID = z.node2ID

Figure 1. *Left side:* The "birthday paradox", the fact that in a group of only 14 people the chance that at least two people have the same birthday is greater than 20 %, modelled with Cool Modes and the *Stochastic Experiments* visual language. *Right side: Condition Tree* for the Calendar Urn. The small boxes (Finished.html, NoConnection.html, NoDrawNode.html) beneath the constraints represent the output that should be provided if this constraint is fulfilled (lighter box) or not (darker boxes) .

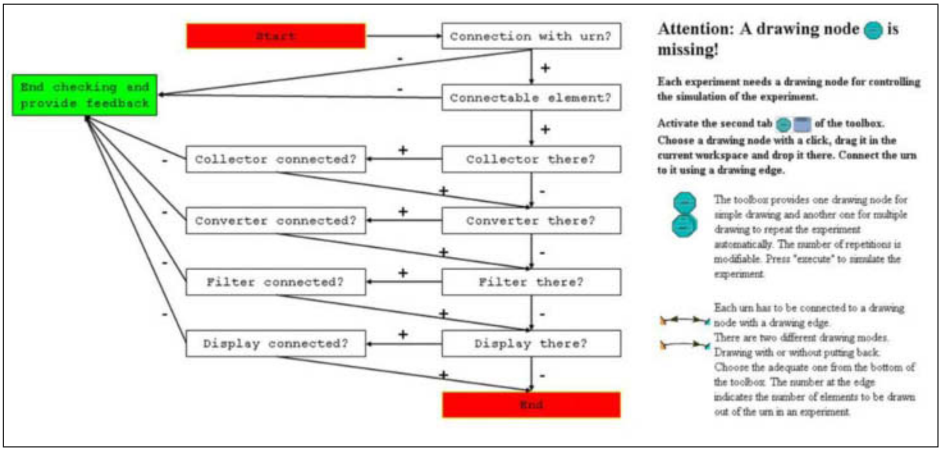
functions depending on its location and connection to other objects. Feedback restricted to 1:1 assignment of an object to a fixed help text is likely to be too unspecific. In this paper, we show how a modular constraint checking system (MCC) can provide advanced help here, taking into account the *context* of the element for which help is required.

2. Help for the *Stochastic Experiments* Plug-In

As an example for interactive help in a modelling process involving visual languages, we now describe the help system for the *Stochastic Experiments* Plug-In for the Cool Modes framework [3]. This visual language provides graphical elements for modelling classic stochastic experiments. Fig. 1 (left) shows an example, which allows students to investigate the famous “birthday paradox”. A control element (b) draws dates from an urn (a). The data is stored in collectors (c, g) and is visualized in display elements (d, f, h). The filter (e) executes automatic analysis processes and enables the computation of the probability (f). The experiment can simulate thousands of groups of arbitrary size. The user can investigate, how many of these groups have members with identical birthdays. Exploring such experiments can build, expand and test student’s knowledge about stochastics. Unfortunately, the *Stochastic Experiments* language is very complex, so it is quite complicated to build models like the one in fig. 1 (left) from scratch. The language contains more than 20 different object primitives, most of which can be used in different contexts. Although already built models are easy to use, students need much help in building them on their own.

The help system we implemented works as follows: The user can request help for any visual object of the *Stochastic Experiments* language. Using this object as a starting point, a checking mechanism analyzes *the whole graph structure* and gives help closely related to the current situation. We do, however, *not* model any knowledge about stochastics in the checking mechanism. Instead, we concentrate on generic parameters like classnames of objects and connections between them (cf. “Semantic Illusion” [4] and “Pseudo Tutoring” [2]). For each object of the *Stochastic Experiments* language, we provide constraint tree(s). The fulfillment of subsets of these constraints trigger the presentation of specialized help files. Fig. 1 (right) shows such a tree for an object representing an “urn” (a in fig. 1 (left)) the user can draw random dates from. In this simple example, there are only three different states to distinguish. For each of the three situations, a specialized help file is presented to the user upon matching (cf. fig. 2 (right) for an example of such a file).

A more complex example is shown in fig. 2 (left). The figure shows a graph that gives an overview about the help system for the *Drawing Node*, the central element controlling the simulation process of the *Stochastic Experiments* language. For this node, there are too many states to be observed to represent them in a single Condition Tree. To handle such complex cases, we introduced a sequencing mechanism, which enables testing relevant states one after another: Each node in the graph in fig. 2 (left) represents a complete constraint tree like the one in fig. 1 (right). When help is requested, this “meta graph” is traversed. Whenever a node is reached, the check represented by the Condition Tree belonging to this node is executed. Depending on the results of this check (whether it is successful or not) the traversion goes on. Only the total of all 10 condition trees in fig. 2 can handle the complex situation recognition needed for this node.



Attention: A drawing node ● is missing!

Each experiment needs a drawing node for controlling the simulation of the experiment.

Activate the second tab of the toolbox. Choose a drawing node with a click, drag it in the current workspace and drop it there. Connect the urn to it using a drawing edge.

The toolbox provides one drawing node for simple drawing and another one for multiple drawing to repeat the experiment automatically. The number of repetitions is modifiable. Press "execute" to simulate the experiment.

Each urn has to be connected to a drawing node with a drawing edge. There are two different drawing modes. Drawing with or without putting back. Choose the adequate one from the bottom of the toolbox. The number at the edge indicates the number of elements to be drawn out of the urn in an experiment.

Figure 2. *Left side:* Overview of the Drawing Node help. Each node in the graph represents a Condition Tree. *Right side:* Feedback example: This feedback presented to the user if there is an urn , but no drawing node in an experiment.

3. Summary

The approach presented in this paper shows how situation dependent help for visual languages can be provided by analyzing the context graph structure of the object for which help is needed. Some advantages of the presented technique are:

- The configuration files that contain the constraints and the HTML files with the help given to the user can be developed and distributed independently of each other and of the visual language.
- This help framework can be used for all the visual languages of the Cool Modes environment. Providing help for a new Plug-In only means building appropriate configuration files. This can be done by domain experts using visual interfaces. (System *users* are able to extend the system, cf. [5])

In the future, we will evaluate the helpfulness of this type of help system compared to systems that only present fixed (i.e, not situation-dependent) help.

References

[1] N. Pinkwart. A plug-in architecture for graph based collaborative modeling systems. In *Proceedings of the AIED 2003*, pp. 535–536.

[2] K. Koedinger, V. Aleven, N. Heffernan. Toward a rapid development environment for cognitive tutors. In *Proceedings of the AIED 2003*, pp. 455–457.

[3] A. Lingnau, M. Kuhn, A. Harrer, D. Hofmann, M. Fendrich and U. Hoppe. Enriching traditional classroom scenarios. In V. Devedzic et al., *Advanced Learning Technologies*, Los Alamitos, 2003.

[4] K. Herrmann, H.U. Hoppe, N. Pinkwart. A checking mechanism for visual language environments. In *Proceedings of the AIED 2003*, pp. 97–104.

[5] G. Fischer. Reflective practitioners and unselfconscious cultures of design. In *Proceedings of the CHI 2004*, 2004.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Knowledge Extraction and Analysis on Collaborative Interaction

Ronghuai HUANG, Huanglingzi LIU
Institute of Knowledge Science & Engineering, Beijing Normal University,
19 Xinjiekouwai St., Beijing, 100875, China

Abstract. E-learning is popularized so fast and Collaborative Learning (CL) becomes so important an instructional strategy. There are huge Group Session (GS) texts needed to be analyzed to evaluate CL, thus the automatic or semi-automatic methods of analyzing the GS texts become very important.

In this paper we present a method called *Interaction Analysis depended on Knowledge Extraction* (IAKE) to analyze collaborative learning by extracting knowledge from the GS texts. This method is based on a GS text analysis approach we proposed it as *Theme based Knowledge Extraction* (TKE).

1. Introduction

Since e-learning becomes more and more popularized, there are huge group session texts of web-based cooperatively learning to be tackled, so that traditional evaluation approaches are not sufficient to tackle the evaluation of web-based collaborative learning.

To analyze the GS texts, we proposed an approach called *Theme based Knowledge Extraction* (TKE). The first step is to extract various “Concepts” from the GS texts. The second step is to generate tree of relationship of concepts by data mining. The third step is to visualize and to model themes.

Based on the Vygotsky’s *Activity Theory*, Huang et al. [1] presented the TAP^2 model for analysis on collaborative interaction. It consists of three dimensions: *Themes Conversion* reflects the shared knowledge constructed or used during the process of problem solving or task completing; *Affective Change* describes the whole emotional relationship and the change of participants’ affective states; *Process Pattern* describes the inter-dynamic strategy for a specific task with the abstract communicative steps.

2. Knowledge Extraction

The GS texts are semi-structured with the unit of group session, and each group session consists of a list of speech given by various roles (group members). Thus, we are not concerned with providing analysis of documents but rather of a subset of the textual database viewed as whole. At the same time, we focus on technologies that may also be used to discover concepts, rules, and relations between separate categories. Thus, *Knowledge Extraction* here is a text version of generalized data mining, and it consists of *Natural Language Processing* (NLP) to extract concepts from each piece of text, statistical analysis to find interesting patterns among the concepts, visualization and modeling.

3. Theme-based Knowledge Extraction

Our textual databases are similar to ones that Nasukawa [2] treated, because the textual databases consist of a list of speech, which are given by various people. However, our textual databases are more structured and more logic. Firstly, the GS texts can be treated in unit of group, and each group session consists of a list of speech given by various group members. Secondly, in each group session, all the members will talk about on one or some specified themes, and the responsive speech is high relative and logic instead of unbending.

We created a method of knowledge extraction called *Theme-based Knowledge Extraction* (TKE), it is shown as in Figure 1.

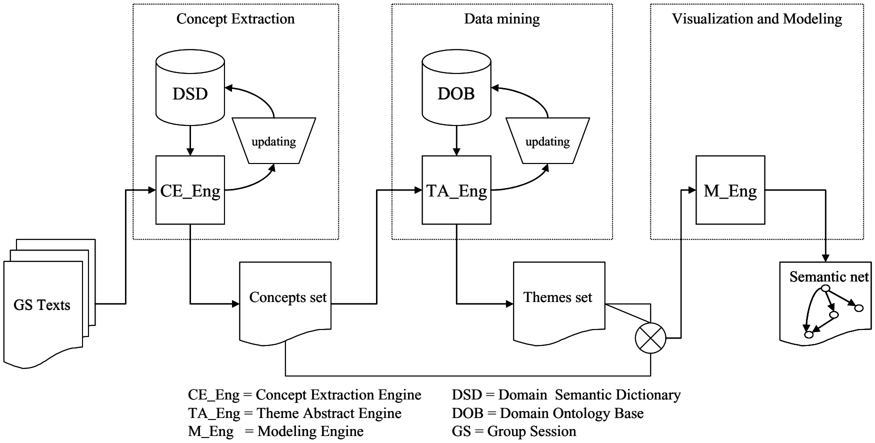


Figure 1 Theme-based Knowledge Extraction

Just like Nasukawa and Nagano suggested in [2], we also use the term “Concept” as a representation of the textual content in order to distinguish it from a simple keyword with surface expression.

There are two main issues in representation of textual contents. The first issue is the polysemy and the synonymity of natural language. The second issue is the differences between Chinese and English, written Chinese texts lack explicit delimiters between words to indicate the boundaries.

In order to deal with the above problems, we create semantic dictionaries, which are called *Domain Semantic Dictionary* (DSD), for different treatment such as *Themes Change, Affective Change and Process Pattern*. A specified analysis goal only needs to deal with the specific words and characters, so the construction on this kind of dictionary becomes comparatively easy. The semantic dictionary is the base to create a special extraction engine.

Concept Extraction Engine (CE_Eng) is an intelligent parser, which scans each GS text to find the concepts matched to the specific DSD, as well as to pick up some similar concepts to be confirmed by user. In addition, it records the concept's position parameters such as role tag and time tag, and stores them to a database *Concepts Set*.

Themes Abstracting Engine (TA_Eng) runs in four steps. The first step is to extract the relationships from the Concepts Set. The second step is to find similar relationship and related concepts to match. The third step is to generate the relationship tree, which is stored in the *Themes Set* as a database. The forth step is to add relationship to the ontology base, which is called *Domain Ontology Base* (DOB) because of being used for the specific goal and the specified contents.

Ontology means terms used in a specific domain, the definition of relationships among the terms, and the expression of the relationships in a hierarchical structure. We suggest a method of constructing ontology semi-automatically from the Concepts Set. Figure 2 shows how to extract terminology and analyze its structure to obtain a hierarchical structure and to add extracted relationships to ontology. These functions are embodied in TA_Eng.

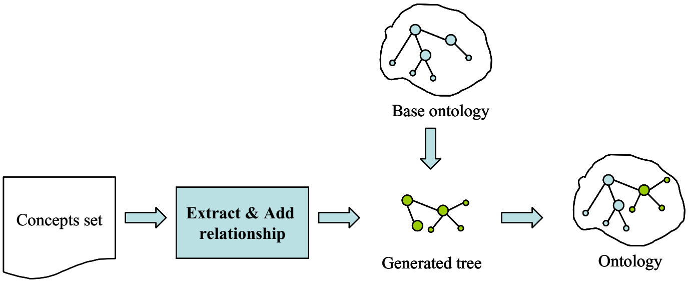


Figure 2 Ontology Construction

The Interaction Analysis depended on Knowledge Extraction (IAKE) consists of three approaches for corresponding dimensions such as themes conversion, affection change, and process route. All three approaches use the same CE_Eng to extract the "concepts" based on Theme's DSD, Affection's DSD and Process Pattern's DSD respectively.

4. Concluding remarks

We have developed an approach of theme-based knowledge extraction to discover knowledge from very large amounts of semi-structured textual data with specified goals. Based on this approach we have developed an interaction analysis method to discover knowledge from group sessions in three dimensions such as themes conversion, affective change and process pattern, as well as to evaluate the collaborative learning process.

References

[1] Huang, R., Liu, H., & Zhu, L. (2005). A Research Toward Collaborative Knowledge Building Through Interaction Analysis. *Open Education Research (in Chinese)*, Vol.11, No.2, pp31-37(2005).

[2] T. Nasukawa, T. Nagano, "Text analysis and knowledge mining system", IBM Systems Journal 40, No 4, pp.967-84(2001).

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Enriching Classroom Scenarios with Tagged Objects

Marc Jansen, Björn Eisen, Ulrich Hoppe
University of Duisburg-Essen
Faculty of Engineering
Institute for Computer Science and Interactive Systems
47048 Duisburg, Germany

Abstract. Over the last recent years, the technology of RF-ID tags enriched a lot of different scenarios. Already at the very beginning, this technology was considered to be useful in the area of CSCL (Computer Supported Collaborative Learning). This paper describes an approach that uses RF-ID tags in order to store information about objects used in a physical simulation environment. Hereby, the stored information and therefore the RF-ID tags help to bridge the gap between a physical environment and a virtual setting in a collaborative modelling and design system.

Introduction

The rapid development of smart devices provides new possibilities for CSCL environments. Milrad et al. described in [1] how smart mobile devices could enrich learning scenarios. Furthermore, the usage of currently growing techniques like RF-ID tags or touch sensitive boards allow for an even broader enrichment of learning scenarios. Eden explains in [2] why it is important that face-to-face learning scenarios are enriched with physical objects. He argues that especially for open-ended learning tasks, the combination of physical and virtual objects are useful in order to on the one hand provide physical objects that can act as a media in a face-to-face setting and on the other hand to be able to do complex calculations or simulations in the virtual world. Eden states that this combination allows to provide "*objects to think with*". While Eden concentrates on touch sensitive boards, Sugimoto [3] moves a step forward by also providing physical objects with RF-ID tags in order to be able to store information about the objects. Sugimoto utilizes the idea of placing several RF-ID scanners directly at the board in order to be able to either calculate the location of an object and to receive information, about the object. Both, Eden and Sugimoto used their environments to provide planning tasks to the learner. In contrast to that, our scenario provides, while using either touch sensitive boards and RF-ID tags, a physical and a virtual modelling and simulation space. In our scenario, we provide students with the task to model the three bodies problem, a well known problem that tries to explain e.g. the movement of stars. Therefore, the students get the possibility to place certain polystyrene balls on a touch sensitive board. Each of these balls has its own RF-ID tag which allows to recognize the ball and to receive certain information about it. In the following, we will describe the different parts of the scenario and how they interact with each other.

1. A Modelling Environment for Universe Animation

The environment is divided into two parts. On the one hand we used the collaborative learning and modelling application Cool Modes [4] that provides a visual modelling language for universe settings/constellations, with planets, shadows and orbits. On the other hand we have a physical representation which is based upon an RFID-kit, a touchboard and a computer. These two parts are connected by an application that is able to connect itself to a collaborative Cool Modes session. After orchestrating a certain situation, the students have the possibility to start a simulation in the virtual world that shows how the planets in the current physical setting would move around each other. To start the animation, the students can either use the touchboard or they can start the animation from the virtual environment. To change certain behaviour like the direction of the movement, the student can use gestures on the touchboard which indicates the behaviour.

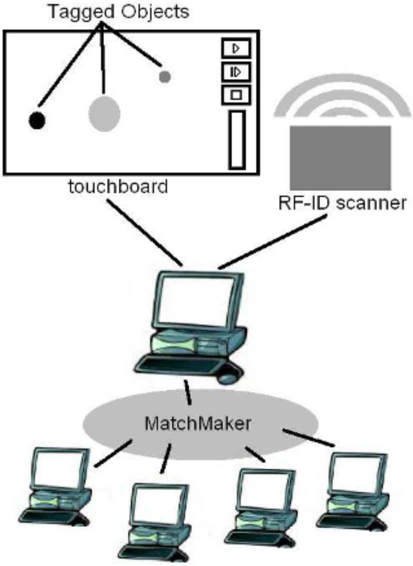


Figure 1 shows the technical architecture of the scenario. In the upper left corner of this figure, the touchboard with three different tagged objects can be seen. The objects are recognized by the RF-ID scanner in the upper right corner. The PC that is directly connected with the touchboard and the communication server, synchronizes the physical world with the virtual one. The several clients that can be seen at the buttom of figure 1 run the Cool Modes applications with the virtual world and are therefore used for the animation.

2. Scenario Usage

Before placing a new object (a sun, a planet or a moon) in the physical universe, it has to be registered by the RFID-scanner. As long as the object is unknown, which basically means that the related RF-ID tag is still empty, any of the connected Cool Modes instances can be used to define its attributes, e.g. the name, the size of the object, its weight, etc. After a valid registration, the object can be placed on the touchboard. To move the smart object on the touchboard, one has to push it down lightly and move it around. Of course the position gets update in the virtual world according to this movement. If a user wants to remove an object, he has to re-register the object with the RF-ID scanner without placing it somewhere on the touchboard. To define the center in a simulated universe where another planet should move around, the user has to drag a circle, with the centerplanet in the center, starting near the planet that should move around the centerplanet. The movement of a moon around a planet is defined accordingly. One important aspect in the usage of the scenario is the fact that always the object that was detected last has to be set on the board as the next object as long as it is not the task to delete the object. Therefore, the creation of a modelled universe will end up in a serialized task of adding objects to the universe and changing their attitudes and behaviour.

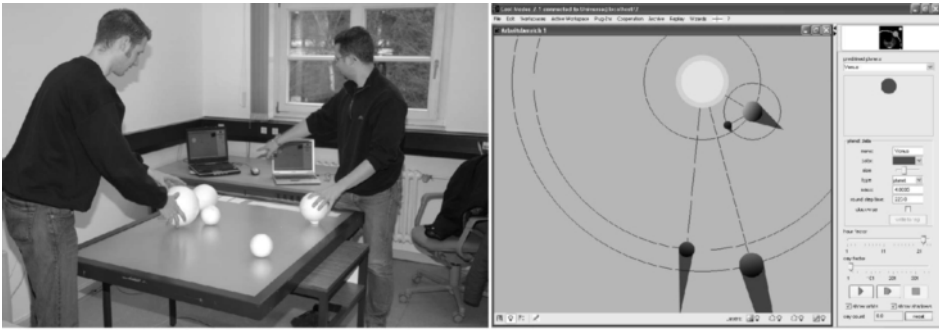


Figure 2 - Two students working on the physical setting and the according virtual representation

Figure 2 shows two students cooperating in the scenario. Additionally, a screenshot of the Cool Modes application with the virtual universe created by the two students can be seen.

3. Future Work

The work presented in this paper is still preliminary in the sense that we have not yet made usability tests with students in schools. Therefore, one aspect of our future work will be to use the presented scenario in schools. Another aspect will be to explore further possibilities to enrich learning scenarios with RF-ID tags and touch sensitive boards. Additionally, the learning scenario will be used in combination with other scenarios in the so-called “life in space” domain. Examples of other scenarios that might be integrated with the presented scenario are lunar cartography, where students should measure distances and heights in order to be able to build a map of the moon; or another scenario, in which students should define parameters for a biosphere in order to be able to grow plants in space.

References

[1] Marcelo Milrad, Ulrich Hoppe, Joshua Gottdenker, Marc Jansen (2004). Exploring the Use of Mobile Devices to Facilitate Educational Interoperability around around Digitally Enhanced Experiments. In: Jeremy Rochelle, Tak-Wai Chan, Kinshuk, Stephen J. H. Yang (eds). Proceedings of the 2nd IEEE International Workshop on Wireless and Mobile Technologies in Education, WMTE 2004, Los Alamitos, California (USA), pp 182-186

[2] Hal Eden (2002). Getting in on the (Inter)Action: Exploring Affordances for Collaborative Learning in a Context of Informed Participation, In Proceedings of CSCL '2002; G. Stahl, Ed.; Boulder, CO, 2002; pp 399-407.

[3] Sugimoto, M., Kunsunoki, F., & Hashizume, H. (2000, December 2-6, 2000). Supporting face-to-face group activities with a sensor-embedded board. Paper presented at the Computer Supported Cooperative Work, Philidelphia, Pennsylvania

[4] Pinkwart, N. (2003). A Plug-In Architecture for Graph Based Collaborative Modeling Systems. In U. Hoppe, F. Verdejo & J. Kay (eds.): Shaping the Future of Learning through Intelligent Technologies. Proceedings of the 11th Conference on Artificial Intelligence in Education, pp. 535-536. Amsterdam, IOS Press.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Testing the Effectiveness of the Leopard Tutor under Experimental Conditions

Ray KEMP, Elisabeth TODD, Rosemary KRSINICH
Institute of Information Sciences and Technology, Massey University, Palmerston North, New Zealand

Abstract. The Leopard Tutor is a piece of software that has been developed to help students learn OO programming. In order to test its effectiveness we ran a course for novice programmers on Java. The participants had a similar background, with little or no knowledge of the language. Half the group used the Leopard Tutor to help them understand the basic concepts and the other half were taught in a traditional manner. In a four hour test administered at the end of the course, the group who had used the Leopard Tutor performed significantly better than the control group.

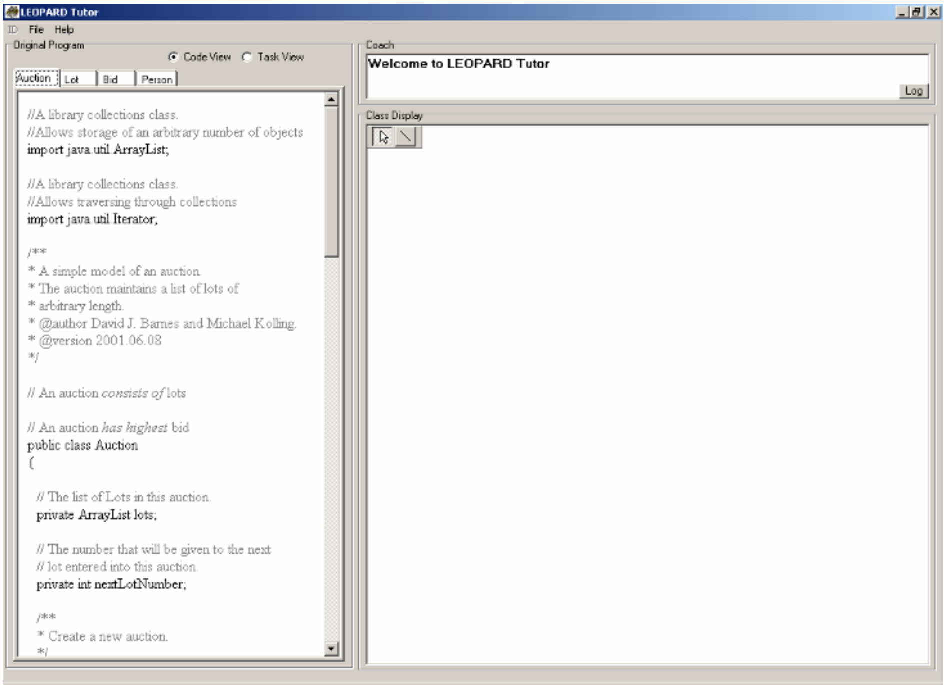
Introduction

Students have problems bridging the gap between reading OO programs and writing their own. The Leopard Tutor (LT) [1] is a piece of software to help them with this transition. Students using LT are challenged to produce both *code level* class diagrams (denoting how the code elements relate to one another), and *task level* ones (showing how the classes denote the real world problem). This helps them to understand the relationship between the real world model of the system, and the code level description in Java.

Figure 1 shows the opening screen from LT. On the left hand side is the program that is supplied by the teacher. The larger pane on the right hand side will contain the corresponding class diagram constructed by the user. The small pane on the upper right hand side will contain feedback from the coach.

Typically, the first program presented will be simple with a single class and a small number of attributes and methods. The student is then challenged to identify these and produce a class diagram in the large pane on the right hand side. LT aids this process by allowing the user to select lines of code from the program and drag them across. Initially, a class box must be created. In order to do this the user must recognize the class heading in the program, select it, and drag it across to the diagram pane. Here, it is converted by LT into a class diagram box, with partitions for attributes, constructor(s) and methods. Next, the user must recognize and select the attribute, constructor and method definitions and drag those across to the appropriate compartments of the class box. Figure 2 shows a screen from LT, where the student is part way through the process for a more complex program with several classes.

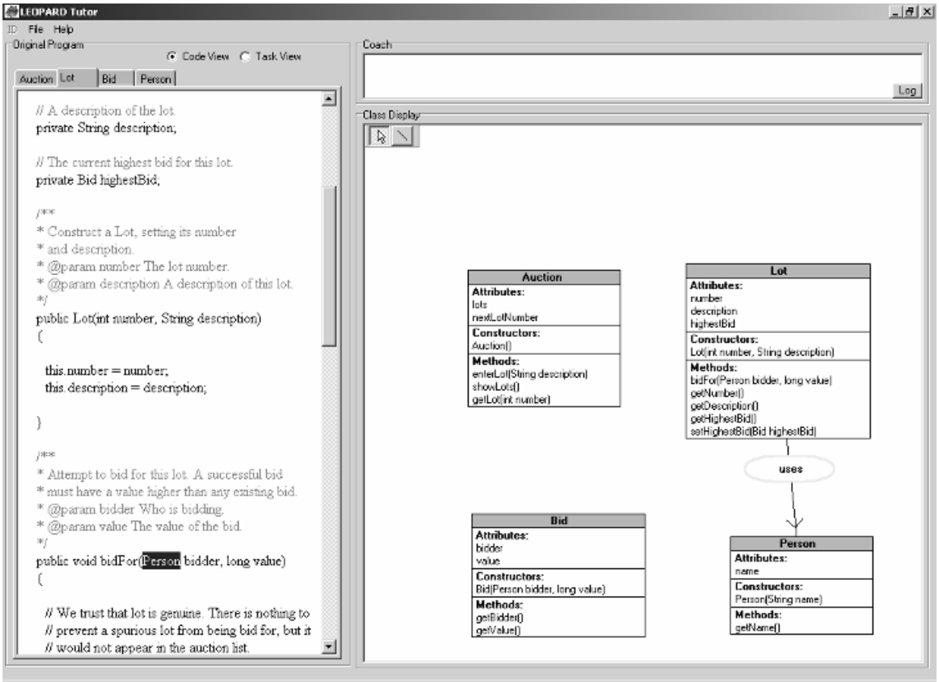
In addition to producing this code level representation of the program, students are challenged to produce a task level one. The default display as shown in Figure 1 has the code level radio button selected, and, correspondingly the code is highlighted with the comments greyed out. When the task view radio button is selected, the comments are highlighted and the code greyed out. The comments are regarded as *advance organizers* [2, 3] and can be used to build up a task level class diagram of the program.



Welcome to LEOPARD Tutor

```
//A library collections class.
//Allows storage of an arbitrary number of objects
import java.util.ArrayList;

//A library collections class.
//Allows traversing through collections
import java.util.Iterator;

/**
* A simple model of an auction.
* The auction maintains a list of lots of
* arbitrary length.
* @author David J. Barnes and Michael Kolling.
* @version 2001.06.08
*/

// An auction consists of lots

// An auction has highest bid
public class Auction
{

  // The list of Lots in this auction.
  private ArrayList lots;

  // The number that will be given to the next
  // lot entered into this auction.
  private int nextLotNumber;

  /**
  * Create a new auction.
  */
```

Figure 1. Opening Screen of Leopard System



```
  // A description of the lot.
  private String description;

  // The current highest bid for this lot.
  private Bid highestBid;

  /**
  * Construct a Lot, setting its number
  * and description.
  * @param number The lot number.
  * @param description A description of this lot.
  */
  public Lot(int number, String description)
  {

    this.number = number;
    this.description = description;

  }

  /**
  * Attempt to bid for this lot. A successful bid
  * must have a value higher than any existing bid.
  * @param bidder Who is bidding.
  * @param value The value of the bid.
  */
  public void bidFor(Person bidder, long value)
  {

    // We trust that lot is genuine. There is nothing to
    // prevent a spurious lot from being bid for, but it
    // would not appear in the auction list.
```

```
Auction
Attributes:
lots
nextLotNumber
Constructors:
Auction()
Methods:
enterLot(String description)
showLots()
getLot(int number)

Lot
Attributes:
number
description
highestBid
Constructors:
Lot(int number, String description)
Methods:
bidFor(Person bidder, long value)
getNumber()
getDescription()
getHighestBid()
setHighestBid(Bid highestBid)

Bid
Attributes:
bidder
value
Constructors:
Bid(Person bidder, long value)
Methods:
getBidder()
getValue()

Person
Attributes:
name
Constructors:
Person(String name)
Methods:
getName()
```

Figure 2. The Leopard Tutor Interface Part Way Through a Session

1. The Experiment

We designed an experiment to test the effectiveness of LT when teaching Java to novice programmers with little or no previous knowledge of the language. All the students in the experiment had a similar background. They had completed two elementary courses in computing at Massey. Within these courses, they had studied a procedural event-driven language and had done a little OO analysis and design but had not been taught any Java.

Eighteen volunteers were recruited and were split into two equal groups. The partitioning was carried out by selecting pairs of students of comparable ability (based on their results in the computing courses) and randomly assigning them to one group or the other. This gave us two mixed ability groups each containing nine students.

The course was an intensive one held over a four day period. The groups spent about two thirds of their study time in common classes. For the remainder of the time the treatment group were learning how to use the Leopard Tutor with corresponding exercises, whereas the control group spent this time consolidating their knowledge of the standard material. At the end they were given a two hour written and a two hour practical test.

The material to be taught included elementary programs and terminology, class definitions, object interaction and grouping of objects. BlueJ [4] was used for teaching and practice: a Java system with a user-friendly interface. The package encourages students to think in terms of OO structures, which fits in well with the Leopard Tutor approach of associating class diagrams with code.

2. Results and Analysis

In the post-course tests, an understanding of program comprehension, class diagram construction, tracing and debugging were all examined. The group who used LT performed significantly better at the 5% level for each of the activities except debugging. However, the debugging results were unreliable since most students struggled with the test question on the topic and only a small number answered the question adequately.

The results from our experiment to test the effectiveness of LT were encouraging. Various aspects of student performance seemed to benefit by exposing them to the software. Their comprehension and tracing, in particular, appeared to be improved. We now feel confident that we should include the software as an integral part of our learning programme for Java.

References

[1] R. H. Kemp, E. Todd, and J. Y. Lu, "A Novel Approach to Teaching an Understanding of Programming," in *Artificial Intelligence in Education: Shaping the Future of Learning through Intelligent Technologies*, J. Kay, Ed. Sydney: IOS Press, 2003, pp. 449-451.

[2] F. Détienne, *Software Design - Cognitive Aspects*: Springer, 2002.

[3] D. Ausubel, *The Psychology of Meaningful Verbal Learning*. New York: Grune & Stratton, 1963.

[4] M. Kolling, B. Quig, A. Patterson, and J. Rosenberg, "The BlueJ System and its Pedagogy," *Computer Science Education*, vol. 13, pp. 249-268, 2003.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Setting the Stage for Collaborative Interactions: Exploration of Separate Control of Shared Space

Lucinda KERAWALLA, Darren PEARCE, Jeanette O'CONNOR, Rosemary LUCKIN, Nicola YUILL and Amanda HARRIS
University of Sussex, Falmer, Brighton, BN1 9RH, UK

Abstract: Most educational software available to children is designed for a single user and this, coupled with a shortage of computers in schools, means that pairs or groups of children often share an inappropriate interface which can be detrimental to collaboration. We describe a novel user interface, Separate Control of Shared Space (SCOSS), and present two studies that explore its potential as a tool to resource collaborative interactions. We illustrate how it can be used to allow for equitable control at both input and task levels, and how it visually represents agreement and disagreement which can be used to mediate collaboration about a final solution.

1. Introduction

Much of the software that is used in classrooms is designed for a single user but is often shared between pairs or small groups. Teachers often promote 'sharing' as being a desirable behaviour and encourage children to take turns with the input device, but this often results in co-operation (e.g. [1]) rather than collaboration. Co-operation can be defined as task-sharing, but when this breaks down, there is the potential for one child to dominate the other and to complete the task without conferring with their partner.

Scott, Shoemaker and Inkpen [2] have found that the provision of multiple mice does not improve the likelihood of concurrent interaction between children. We argue that this is because the software interface used in their study allowed only one child to have access to each feature at any one time, thus promoting turn-taking rather than concurrent task activity. Likewise, Benford, Bederson and Akesson et al [3] report that children using KidPad with a mouse each co-operated effectively on task-sharing but that reciprocal discussion was minimal, compared to children who were asked to share a single mouse. We argue that this is a function of the KidPad interface and of the task: the children are given the option of distributing task elements between them, which they complete separately.

Another field of research that addresses the issue of interface design to support collaboration is Computer Supported Collaborative Learning. For example, Suthers [e.g. 4] explores how the design of representational tools can support students' collaborative discourse. We argue that although this interface, and others developed in this field (e.g. [5]), is designed to mediate collaborative interactions, there is the possibility of one student dominating the other by deleting and over-riding their partner's contribution without any discussion.

2. Separate Control of Shared Space: Features

To overcome some of these limitations, our interface, Separate Control of Shared Space (SCOSS), enforces each person to engage with the task. It provides each user with their own space and each child can control only elements within their own space; it is not possible to delete their partner's work, which is what single user interfaces, even with dual control, are unable to do. Dual control of a single user interface results in users taking it in turns to move a single representation of each element on the screen, whereas the SCOSS interface allows for equity at both input and task-process levels which gives each user the potential for becoming engaged with each and every element of the task. However, the SCOSS interface can only provide users with *opportunities* for equity in the task process; the amount exerted is up to individual users. We have also included a 'we agree' feature on the SCOSS interface and this can be adapted to different tasks (at the programming stage) so that agreement can occur at pre-defined stages or upon completion of the whole task.

3. Study One

The main aim of study one was to compare the utility of the SCOSS interface with single control of a single user interface, and dual control of a single user interface. The focus was on determining whether children could use SCOSS as a tool to mediate their collaborative progress through the task. The SCOSS interface was studied in a simple task in which thirty six pairs of 8-9 year olds were asked to estimate the number of sweets in eight containers (e.g. a small box, a large jar).

In condition A the children shared a single-user interface: they shared a single keyboard and their estimates were represented on a single scale. There was one 'we agree' key to be pressed to indicate that the children agreed on their estimates at each stage of the estimation process. In condition B the children used an interface that represented dual input into single user software (as used by [2] and discussed above). The children saw the same interface on the screen as in condition A, but each child had their own set of designated keys. In Condition C the pairs used the SCOSS interface. Each child had their own set of keys (as in B) but they also had their own space in which to work along with their own agree key to indicate when they were in agreement on their final estimate.

Video footage was coded using a scheme that identified whether collaboration was occurring in terms of evidence of 1) justification of opinions/answers, 2) joint understanding, 3) joint agreement, 4) working towards a shared goal, 5) equal opportunity to contribute to the task and 6) equitable opportunity for input. Analysis revealed that there was no difference in the quality of the collaboration across the three conditions: with the exception of 2 pairs of children, all pairs were poor collaborators due to either failing to provide justifications for their estimates (15 pairs), and/or failing to work towards a shared goal (1 pair) and/or one child dominating the proceedings (3 pairs). This indicates that some children are not spontaneously good at collaboration and confirms our hypothesis that SCOSS alone cannot mediate the *quality* of the discussion surrounding decisions. However, there is video evidence of the potential for the SCOSS interface to mediate joint decision-making by making agreement and disagreement visually explicit .

4. Study Two: SCOSS with discussion prompts.

A further study investigated the utility of the SCOSS interface in mediating collaboration between 10 parent and 5/6-year-old child pairs in the home context. This study focused on parents and children completing an activity where they completed weight and mass tasks. 'Frankie's Fruitful Journey' was designed by O'Connor [6] to incorporate the principles of the SCOSS interface that have been discussed above. This software also incorporated the use of discussion prompts to scaffold the quality of the collaboration (see [7] for details).

Observations of the conversations indicated that neither the adult nor the child dominated the decision-making process. All participants made different choices from their partner in the first part of each task indicating that the SCOSS interface supported individual decision making actions and enforced participation. All the adults used the visual representation of agreement and disagreement provided on the interface by physically pointing out differences on the screen to focus the attention of the child.

5. Discussion

These studies have demonstrated the potential for the novel user interface - Separate Control of Shared Space - to mediate collaborative interactions. It is effective in setting the stage for: individual agency at input and task-process levels, representing agreement and disagreement, and mediating eventual agreement. These interface features can be used as a resource by users to mediate their conversation. The addition of discussion prompts effectively scaffolds collaborative conversations so that interchanges are more rich and contain more examples of users explaining and accounting for their decisions. This, in turn, means that there is a higher level of joint understanding between users.

In future research we would like to explore the issue of task dominance more closely. We plan to investigate whether training and practice in both collaborative skills and use of the SCOSS interface would be of benefit to child-child pairs. Furthermore, we have realised the potential of the SCOSS interface as a tool to provide data about individual contributions to the task, which can be used by a learner model which in turn can offer further scaffolding for the collaborative process. This future work will help us to build upon that reported here and enable us to realise the full potential of the SCOSS interface.

References

[1] Dillenbourg P. (1999), What do you mean by collaborative learning? In Dillenbourg P. (Ed), *Collaborative Learning: Cognitive and Computational Approaches*, Pergamon, Amsterdam, pp 1-19

[2] Scott S., Shoemaker G. and Inkpen K. (2000), Towards seamless support of natural collaborative interactions, in *Proceedings of Graphics Interface*, Montreal; Canada May. pp103-110.

[3] Benford, S., Bederson, B., Akesson, K., Bayon, B., Druin, A., Hansson, P., Hourcade, P., Ingram, R., Neale, H., O'Malley, C., Simsarian, K., Stanton, D., Sundblad, Y. & Taxen, G (2000), Designing Storytelling technologies to encourage collaboration between young children. In *Proceedings of CHI 2000*, The Hague, Netherlands, April1-6.

[4] Suthers D. (1999), Representational support for collaborative inquiry, *Proceedings of the 32nd Hawaii International Conference on System Sciences*, January 5-8, Maui; Hawaii.

[5] Dorohonceanu B., Sletterink B. and Marsic I. (2000), A novel user interface for group collaboration, in *Proceedings of 33rd Hawaii International Conference on System Sciences*, January 4-7, Maui; Hawaii.

[6] O'Connor J. (2004), *The Design and Evaluation of Frankie's Fruitful Journey: A Computer-Based Activity to Facilitate Parent-Child Scaffolding as a Method to Support the Child's Understanding of Mass in Key Stage 1*, MSc Dissertation, University of Sussex.

[7] O'Connor J. and Kerawalla L. (2005 forthcoming), The use of discussion prompts to scaffold parent-child collaboration within a computer-based activity, AIED Conference, Amsterdam.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Computer Simulation as an Instructional Technology in AutoTutor

Hyun-Jeong Joyce Kim[1], Art Graesser[2], Tanner Jackson[2]
Andrew Olney[2], and Patrick Chipman[2]
[1]*Department of Psychology, Rhodes College, 2000 N. Parkway, Memphis, TN 38112,*
[2]*Department of Psychology, University of Memphis, Memphis, TN 38152*

Abstract. We explored the impact on learning of interactive simulations that were coordinated with AutoTutor, a learning environment that helps students by holding a conversation in natural language. We randomly assigned 132 college students to one of three conditions: AutoTutor without simulations, AutoTutor with simulations, and a Monte Carlo AutoTutor that randomly generated dialogue moves. A pretest-posttest design was used to measure learning gains, as measured by objective multiple choice questions. All versions of AutoTutor were successful in promoting learning. The Monte Carlo AutoTutor produced significantly lower gains than the interactive simulation version for higher knowledge learners, and the direction of the three means were in the predicted direction. Improved simulation dialogues, modeling of good simulation manipulation strategies, and faster display of simulations are expected to enhance learning in future versions of AutoTutor.

1. Background

Constructivist views of learning emphasize the importance of the learner's active exploration and knowledge construction, rather than mere information transmission. One recent constructivist method to stimulate students' cognitive activity and facilitate their active construction of knowledge is a simulation-based environment. A number of early studies investigated the use of simulations as an instructional technology within computer-based learning environments [1]. Somewhat surprisingly, a mea-analysis of 93 studies conducted by Dekker and Donatti found mixed results on the effects of simulations [2]. The question arises as to why their evidence is inconclusive. Potential flaws in the studies might include poorly designed simulations, speed of display, difficulty of subject matter, and flexibility of user control.

Although there is a large body of research on computer simulations, researchers have not yet conducted research on the impact of simulations when it is coupled with dialogues. This motivated us to develop a tutoring system that constructs dialogues during the simulations, with guidance on how to use the simulations and suggestions on what to do next. The long-term goal is to create a computerized tutoring system that can select intelligent dialogue moves that can effectively guide the learner through a simulated environment [3].

2. AutoTutor

AutoTutor is a web-based computer tutor that holds conversations with students in natural language, that simulates dialogues that human tutors typically use, and that teaches students conceptual physics and computer literacy [4]. AutoTutor has an animated agent with a synthesized speech, facial expressions, and gestures. Recently AutoTutor has combined mixed-initiative dialogue with interactive simulations [3].

The simulation environment of AutoTutor is an embedded 3-D world with a set of parameters (e.g., speed of objects, distance between objects). The simulation environment was designed so that learners can use both slider and toggle controls to alter the simulations and run and rerun a simulation as many times as they desire. This allows users to practice at their leisure, and to learn at their own pace. Along with the simulations, there is corresponding tutorial dialogue which was designed to scaffold the learning process [4]. While simulations are running, AutoTutor stops at various points to portray relevant physics principles or to rectify misconceptions.

3. Present Empirical Study

The present study examined the efficacy of simulations and dialogues in AutoTutor. We compared three tutors: AutoTutor with conversation only, AutoTutor with interactive simulations, and AutoTutor with conversation generated by Monte Carlo generation of dialogue moves. AutoTutor with conversation only (AT) is the typical style of interaction from previous studies [4]. AutoTutor with simulations (AT-Sim) is the same as the conversation-only version except that it has the added component of simulations and the corresponding simulation dialogues. The third tutoring condition, the Monte Carlo tutor (AT-MC), does not use intelligent selection of the next dialogue move when trying to get the student to articulate a particular sentence-size expectation. Instead, a large number of dialogue observations were compiled from previous AutoTutor studies (associated with a particular expectation E) and the resulting distributions were used to determine the selection probability of different dialogue moves. AT-MC serves as a content control condition for the AT condition, but does not tailor particular dialogue moves to particular learners.

The current study assessed the effectiveness of the three different versions of AutoTutor and their respective impact(s) on learning. We predicted that learning gains would increase with the level of tutoring sophistication: AT-Sim > AT > AT-MC.

3.1 Methods

The participants were 132 students from Rhodes College and University of Memphis who were paid for their participation. The experiment consisted of three phases: a pre-test phase, a learning phase, and a post-test phase. During the pre-test phase, all participants were administered 26 multiple choice questions (pulled from the Force Concept Inventory). During the learning phase, participants answered four physics problems while interacting with one of the AutoTutors. The post-test phase consisted of a different set of 26 multiple choice questions (counterbalanced with the pre-test), and a user perception survey. The experiment took approximately two hours to complete.

3.2 Results and Discussion

We compared the three different tutors using four outcome indices: pre-test, post-test, simple learning gains (post test – pre test), and proportional learning gains [(post-test proportion – pre-test proportion) / (1 – pre-test proportion)]. The pre- and post-tests were converted to proportion correct scores.

There were no significant differences in pre-test scores between the tutoring conditions. Overall, we found that all the versions of AutoTutor produced significant learning gains; posttest scores (*M* =.60, *SD* =.18) were significantly better than pretest

Table 1

Means and Standard Deviations for Learning Measures.

Tutor conditions	Pretest	Posttest	Simple Learning Gains	Proportional Learning Gains
AT-Simulation	.459 (.18)	.633 (.17)	.174 (.14)	.309 (.25)
AT	.442 (.20)	.589 (.19)	.147 (.15)	.271 (.25)
AT-Monte Carlo	.464 (.20)	.582 (.18)	.118 (.13)	.237 (.27)

scores (M = .46, SD = .19), F(1, 129)= 141.17, p < .001, effect-size = 0.74. Although no other effects were significant, the data trend supported the predictions: AT-Sim > AT > AT-MC. Table 1 shows the cell means and SD's in the analyses.

A 2 (pre vs. posttest scores) x 3 (three tutor conditions) x 2 (low vs. high knowledge) ANOVA showed a significant interaction between test scores and a domain knowledge, F(1, 90) = 23.44, p < .01. The difference between pre and post scores was significantly greater for students with low domain knowledge than those with high domain knowledge. Thus, students with low knowledge benefited more from AutoTutor than those with a high knowledge. More interestingly, when we used participants whose pre-test scores were greater than .5, we found a significant difference in the simple learning gains between AT-Sim and AT-MC, F(1, 28) = 4.19, p< .05. AT-Sim produced significantly higher learning gains than AT-MC. This indicates that Monte Carlo tutor might inhibit learning for high knowledge participants and learning gains might suffer without adaptive dialogues.

We are currently in the process of revising the simulation dialogues and improving the simulation environments. Improved simulation dialogues, faster display of simulations, and modeling of effective learning with simulations might ultimately help students to learn deeply about abstract physics concepts. Interactive simulations will hopefully show some promise as a new medium for dialogue scaffolding, creating an immersive environment in which the learner and tutor can interact.

Acknowledgements

The research on AutoTutor was supported by the National Science Foundation (SBR 9720314, REC 0106965, REC 0126265, ITR 0325428) and the DoD Multidisciplinary University Research Initiative (MURI) administered by ONR under grant N00014-00-1-0600 (visit http://www.autotutor.org).. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DoD, ONR, or NSF.

References

[1] Choi, B., & Gennaro, E. (1987). The effectiveness of using computer simulated experiments on junior high students' understanding of the volume displacement concept. *Journal of Research in Science Teaching, 24(6),* 539-552.

[2] Dekker, J., & Donatti, S. (1981). The integration of research studies on the use of simulation as an instructional strategy. *Journal of Educational Research, 74(6),* 424-427.

[3] Graesser, A.C., Chipman, P., Haynes, B., & Olney, A. (in press). *AutoTutor: An intelligent-tutoring system with mixed-initiative dialogue. IEEE Transactions in Education.*

[4] Graesser, A.C., Lu, S., Jackson, G.T., Mitchell, H., Ventura, M., Olney, A., & Louwerse, M.M. (2004). AutoTutor: A tutor with dialogue in natural language. *Behavioral Research Methods, Instruments, and Computers, 36*, 180-193.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Developing Teaching Aids for Distance Education

Jihie Kim, Carole Beal, and Zeeshan Maqbool
USC/Information Sciences Institute,
Marina del Rey, CA 90292, USA

Abstract. As web-enhanced courses become more successful, they put considerable burdens on instructors and teaching assistants. We present our work on developing software tools to support instructors by A) semi automatic grading of discussions and B) creating instructional tools that handle many student requests. We are using knowledge-based techniques in modelling course components, student queries, and relations between them. The results from our initial analysis in developing such tools are also presented.

Introduction

Web-enhanced courses and distance education courses are becoming increasingly popular. Such courses make class materials easily accessible to remote students, and increase the availability of instructors to students beyond the traditional classroom. However, as such courses become more successful, their enrollments increase, and the heavier on-line interaction places considerable burdens on instructors and teaching assistants. Thus, the ultimate success of web-based education is constrained by limited instructor time and availability. At the same time, many routine student queries and on-line activities do not necessarily require instructor or TA intervention. Software tools that can handle some student activities would allow instructors to focus on queries and activities that truly require their attention.

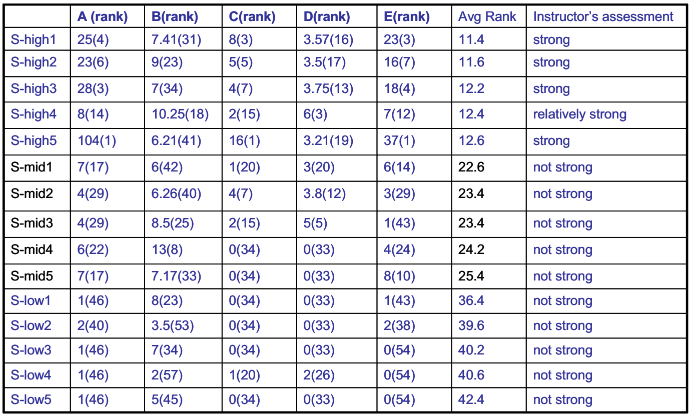
1. Turning quantity into quality: Development and validation of a measure to support automatic assessment of on-line discussion contributions

Engagement in on-line discussions is an important part of student activities in distance education, and instructors often use it to measure each student's contribution to the class. Although it is probably not feasible or pedagogically appropriate to completely automate the grading process, we are developing approaches to semi-automate some of the work.

There has been some prototype measures of discussion quality that relies on the quantity of discussion contributions [2], which include the number of posted comments and the number of responses that a post elicits from classmates and/or the TA or instructor. We are extending the framework to accommodate various factors. Posts that engage many different students might have a higher probability of being high quality than a post that does not elicit interest from anyone else. If a student was involved in various discussions on different topics, we may infer that he/she has broader interests than a student who contributes to only small number of topics.

We are currently collecting course data from various fields including Psychology, Mechanical Engineering and Computer Science. Here we report an initial analysis of a graduate-level Computer Science class on Advanced Operating Systems held in Fall 2003. The course had over 80 graduate students enrolled. Students were encouraged to participate in an on-line forum to discuss on general issues as well as course topics. Their participation was reflected in their grades as class participation scores, consisting up to

10% of the final grade. Table 1 presents a part of our results, showing ranks from three different groups: 5 students with highest ranks, 5 students with middle ranks, and 5 students with lowest ranks. The ranks are computed based on the following factors: A) total number of messages sent, B) average length of the threads where the student participated, C) total number of threads initiated by the student, D) average number of other students involved in the threads that the student initiated, E) total number of different threads where the student participated. The last column shows qualitative assessment of student participation by the instructor.



	A (rank)	B(rank)	C(rank)	D(rank)	E(rank)	Avg Rank	Instructor's assessment
S-high1	25(4)	7.41(31)	8(3)	3.57(16)	23(3)	11.4	strong
S-high2	23(6)	9(23)	5(5)	3.5(17)	16(7)	11.6	strong
S-high3	28(3)	7(34)	4(7)	3.75(13)	18(4)	12.2	strong
S-high4	8(14)	10.25(18)	2(15)	6(3)	7(12)	12.4	relatively strong
S-high5	104(1)	6.21(41)	16(1)	3.21(19)	37(1)	12.6	strong
S-mid1	7(17)	6(42)	1(20)	3(20)	6(14)	22.6	not strong
S-mid2	4(29)	6.26(40)	4(7)	3.8(12)	3(29)	23.4	not strong
S-mid3	4(29)	8.5(25)	2(15)	5(5)	1(43)	23.4	not strong
S-mid4	6(22)	13(8)	0(34)	0(33)	4(24)	24.2	not strong
S-mid5	7(17)	7.17(33)	0(34)	0(33)	8(10)	25.4	not strong
S-low1	1(46)	8(23)	0(34)	0(33)	1(43)	36.4	not strong
S-low2	2(40)	3.5(53)	0(34)	0(33)	2(38)	39.6	not strong
S-low3	1(46)	7(34)	0(34)	0(33)	0(54)	40.2	not strong
S-low4	1(46)	2(57)	1(20)	2(26)	0(54)	40.6	not strong
S-low5	1(46)	5(45)	0(34)	0(33)	0(54)	42.4	not strong

Table 1: Student participations in discussions.

As shown in the table, the instructor agreed that in fact the top 5 students provided strong contributions to the discussions and other students were less strong. Also, we found that there are some correlations between A,C, and E factors. We are currently validating actual correlations between these factors and analyzing other factors that can be potentially useful.

2. Developing instructional tool that semi-automatically answers student queries

The goal of this part of the work is to develop a tool that can handle many of student requests semi-automatically. The tool will seek the instructor's help only when the student needs additional help.

As an initial step, we are focusing on routine queries on general course information, administrative issues on assignments and exams, and other frequently asked questions. Instructors agree that they often spend a significant amount of time although many of them do not actually need their intervention. We are developing 1) a *course ontology* that represents generic components of distance education courses, 2) a *query ontology* that describes types of student queries and requests, and 3) general mappings between the two ontologies, i.e., how a type of query can be addressed by some course components. They are being built as general background knowledge which can support various reasoning capabilities such as classification, verification and knowledge authoring across different courses [1].

Note that these ontologies can include dependencies between different components. For example, participation to the discussion forum is *enabled* when the student knows how to access the forum class. Attendance policy is a *part of* grading policy if class participation grade counts in attendance rate. Figure 1 shows the current ontology we are developing based on the Operating Systems course described above. The left hand side shows the concepts representing the course components. The right hand side shows types of student requests. The actual class structure and its materials are being represented in terms of these concepts and their relations. For example, the course is represented in terms of its syllabus, general information (office hours, exam dates, etc.), distance education network (DEN) relevant information, etc. Each student query is mapped to query types based on the keywords in the message. In the figure the numbers next to query types mean the numbers of actual queries in the class. The lines in the figure highlight mappings between query types and course components.

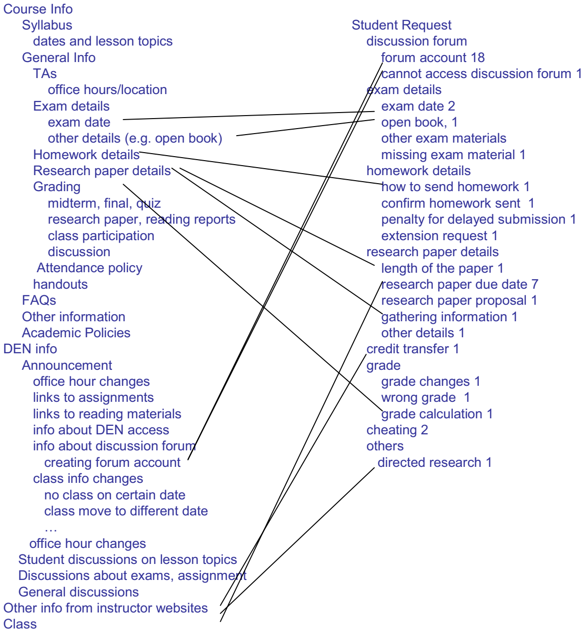


Table-2. ontology of general information about a class and its mapping to student queries

By making these relations explicit, the system can map student queries to relevant course materials efficiently and the results can be sent to the students. When the system cannot find appropriate mappings or the student is not satisfied with the materials sent, the system may bring the case to the instructor's attention. All the interactions between the system and the student will be available to the instructor.

The ontologies enable the system to find answers when simple keyword based search fails.

The following shows an example of such a case. Although the student is asking about message posting and registration, the actual information he needs is how to access his forum account shown below.

Student: I am unable to post message in the Class Discussion. In fact I didn't receive any activation key in e-mail upon registration. Could you please suggest me a way out ?

Course info: Your forum accounts have been created. Your username is the the username part of your school e-mail account, e.g., if your e-mail address gbush@school.edu, then your username for the forum is gbush. Your temporary password is ....

If the system simply uses the content of the message, it may retrieve other instructional components such as how to register for DEN to access DEN materials, which does not help the student in this case. In order to provide an appropriate answer, the system needs to know what information will help the student in the given situation, such as discussion forum account enables the student participation in the discussion forum.

The ontology can be also used in assisting the instructor. The system can show how certain answers were derived by tracing the concepts and their relations used during answer generation. The instructor can use the ontology in organizing their instructional materials and the system can check whether there is any missing or duplicate information by checking dependent components. We are planning to extend our ontology to take into account of the history of student activities, making the context of the queries more explicit.

Acknowledgement

We thank Dongho Kim for providing discussion data.

References

[1] Kim, J. and Gil, Y., Knowledge Analysis on Process Models, *Proceedings of IJCAI-2001*.

[2] Shaw, E. Assessing and Scaffolding Collaborative Learning in Online Discussions, *Proceedings of AIEd-2005*.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Who Helps the Helper? A Situated Scaffolding System for Supporting Less Experienced Feedback Givers

[1]Duenpen KOCHAKORNJARUPONG, [1]Paul BRNA, and [2]Paul VICKERS

[1]The SCRE Centre, University of Glasgow, St Andrews Building
11 Eldon Street, Glasgow, G3 6NH, Scotland, UK
[2]School of IET, Northumbria University
Newcastle Upon Tyne, NE2 1XE, England, UK
duenpen@scre.ac.uk, paul.brna@scre.ac.uk, paul.vickers@unn.ac.uk

Abstract: This research emphasizes the construction of feedback pattern. A system called McFeSPA is designed to help inexperienced teaching assistants (TAs)[1] who lack training in how to provide quality feedback. The system employs scaffolding to help the TAs improve quality feedback skill while marking assignments. We have currently been implementing the system with techniques drawn from Artificial Intelligence, cognitive psychology and theories of education. Our next step will entail the examination of the system for both scaffolding turned off to help two TAs give feedback to a group of students and two TAs using the full system with scaffolding.

1. Introduction

The aim of our research is giving intelligent support for feedback givers with the help of feedback patterns [1], situated within the context of marking programming assignments. Although “feedback patterns” have been proposed in the pedagogical patterns project [2], they have not been implemented in ITS & AIED communities [3] to assist novice TAs become experienced teachers. McFeSPA employs some techniques to help teaching and learning based on feedback giving by experienced teachers. Although automated/semi-automated marking assignment systems can help teachers mark programming assignments (e.g. CourseMaster [4]), they don't explicitly scaffold novice TAs learning to give feedback, their main aim being to make marking assignments easier. In order to carry this research out, there are many questions that need to be asked including “How do people learn to give quality feedback” (and what is quality feedback)? “What does the feedback giver need to learn in order to help the learner”?

2. Scaffolding Framework

The scaffolding approach has been selected as appropriate for TAs who, like adults, have little time to learn anything while engaged in marking students scripts. Although the implementation of scaffolding is difficult, scaffolding techniques have been deployed effectively in a number of systems (e.g. Ecolab [5]). We have chosen to work on the problem faced by the TAs in the realistic situation of marking programming assignments

[1] Inexperienced TAs mean novice TAs including novice teachers, novice tutors, and novice lecturers

for large classes and providing feedback on the students' errors. The TAs are likely to be inexperienced in giving feedback even if they have excellent programming skills. Helping TAs learn to mark programming assignments is close to the method of providing cognitive apprenticeship [6], and consists of content, methods, sequencing, and aspects of social learning. We include this in the framework for designing McFeSPA[2]. McFeSPA's architecture is presented in Figure 1. In this paper, we summarize the approach in Table 1.

Element of McFeSPA framework
Content: Two kinds of domain knowledge: about feedback (knowledge of feedback patterns, knowledge of scaffolding, knowledge of quality feedback), the programming domain (knowledge of errors/weaknesses), heuristic knowledge (rules for feedback pattern, rules for providing quality feedback, rules for tutor's hint, and rules for dialogue response)
Methods: An integrated set of cognitive and metacognitive skills through the process of observation and guided and supported practice as well as implementing fading within McFeSPA
Sequencing: Applying the approach/skill of giving quality feedback to any course of assignments marking based on the users' experience
Social Learning: Situated learning (learning to give quality feedback in the situation of marking real assignments) and learning within a culture focused on and defined by expert practices

Table 1 Element of McFeSPA framework

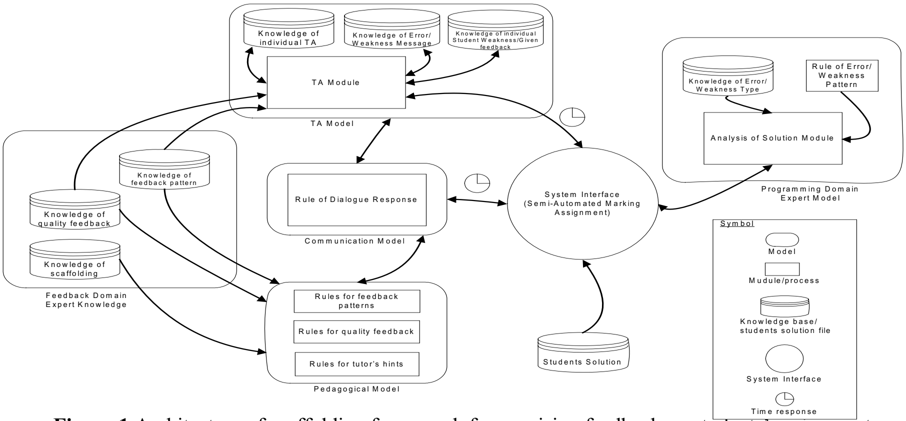

Knowledge of individual TA
Knowledge of Error/ Weakness Message
Knowledge of individual Student Weakness/Given feedback
TA Module
TA Model
Knowledge of Error/ Weakness Type
Rule of Error/ Weakness Pattern
Analysis of Solution Module
Programming Domain Expert Model
Knowledge of feedback pattern
Knowledge of quality feedback
Knowledge of scaffolding
Feedback Domain Expert Knowledge
Rule of Dialogue Response
Communication Model
System Interface (Semi-Automated Marking Assignment)
Rules for feedback patterns
Rules for quality feedback
Rules for tutor's hints
Pedagogical Model
Students Solution
Symbol
Model
Mudule/process
Knowledge base/ students solution file
System Interface
Time response

Figure 1 Architecture of scaffolding framework for provision feedback on students' assignment

As can be seen in Figure 1, The TA receives the student's solution from the interface of the system. Then the system analyses the student's solution based on the error or weakness patterns detected. Thereafter, the system annotates error or weakness patterns and sends this to the TA module. In this stage the system allows the TA to add/update/delete further weakness messages, extending the system. This module will compare each student's weakness with their previous weaknesses and the current weakness in order to help the TA provide appropriate feedback to the student. The TA module stores some information about what the TA does and this module will hold the information which helps TAs to reflect on their work – for example "doesn't do very much" or "doesn't spend a lot of time on reworking the Analysis of solution", and so on. This module depends on monitoring the time taken by the TA, and also employs the knowledge of feedback pattern and the knowledge of quality feedback to help the TA organise the feedback for the student before generating the feedback report to the student. During this process, some information is passed between the Communication module which uses the rules for Dialogue Response and the Pedagogical module which uses the rules for hints, the rules

[2] McFeSPA will run in two modes - scaffolding on or off - this is done for experimental reasons - see later.

for quality feedback, and the rules for feedback patterns. The Pedagogical module utilises three knowledge bases which are for scaffolding, quality feedback, and feedback patterns in order to scaffold the TA to provide quality feedback.

In addition, the design and implementation of McFeSPA includes several forms of scaffold: Functional: the explanation of any components in McFeSPA; Content: five levels of contingent help; and Metacognitive: the hints are designed to help the TA rethink his/her decision, a form of scaffolding for reflection [7]. This latter type of scaffold can help the learner to be aware of his/her own learning through reflection, monitoring, etc. For example, the assessment of understanding "Do I know more/understand better now?"

3. Conclusion, & Future work

From our combination of feedback patterns and our conceptualisation of quality feedback in terms of the five levels of contingent help in McFeSPA, we have hypothesised that McFeSPA could help the TAs learn to give feedback – and could also help the TAs improve their practical feedback skills while fading could promote better help seeking activity. The system is not designed to be a complete solution for supporting the TA e.g. it does not support any interaction between the TA and the student receiving the feedback; nor does it directly support marking assignments even though some error detection is available. Thus, we believe that this research makes a novel contribution to the field of AIED by focusing on how to train people to give quality feedback while marking assignments, in our case, in the context of teaching programming. McFeSPA helps TAs directly, and students indirectly. However, we cannot guarantee that TAs will be happy with the way McFeSPA works given that this depends on its usability which has yet to be determined. After further improvements, our long-term aims include the development of McFeSPA to provide scaffolding for a range of ILEs and also to provide services for web-based systems.

References

[1] Kochakornjarupong, D. and Brna, P. (2003). Towards Scaffolding Feedback Construction: Improving Learning for Student and Marker. In Y.S. Chee, et al. (Eds.), Proceedings of the 11th International Conference on Computers in Education, (ICCE 2003) (pp. 599-600). 2-5 December 2003, Hong Kong.

[2] Eckstein, J., Bergin, J., and Sharp, H. (2002). Patterns for Feedback. In Proceedings of EuroPLoP 2002. Seventh European Conference on Pattern Languages of Programs, Irsee, Germany.

[3] Kumar, V., McCalla, G., and Greer, J. (1999). Helping the Peer Helper. In Proceedings of International Conference on Artificial Intelligence in Education, Le Mans, France.

[4] Higgins, C., Symeonidis, P., and Tsintsifas, A. (2002). The Marking System for CourseMaster. In Proceedings of the 7th Annual Conference on Innovation and technology in Computer Science Education.

[5] Luckin, R. and du Boulay, B. (1999). Ecolab: The Development and Evaluation of a Vygotskian Design Framework. International Journal of Artificial Intelligence in Education, 10, 198-220.

[6] Collins, A., Brown, J.S., and Newman, S.E. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L.B. Resnick (Ed.), Knowing, learning, and instruction: Essays in honor of Robert Glaser (pp. 453-494.). Hillsdale, NJ: Erlbaum.

[7] Jackson, S.L., Krajcik, J., and Soloway, E. (1998). The design of guided learner-adaptable scaffolding in interactive learning environments. In Proceedings of CHI 98, Los Angeles CA.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Realizing Adaptive Questions and Answers for ICALL Systems

Hidenobu Kunichika[*1], Minoru Urushima[*2], Tsukasa Hirashima[*3] and Akira Takeuchi[*2]
[*1] *Dept. of Creation Informatics, Kyushu Institute of Technology, Japan*
[*2] *Dept. of Artificial Intelligence, Kyushu Institute of Technology, Japan*
[*3] *Dept. of Information Engineering, Hiroshima University, Japan*

Abstract. Language training systems that provide learners adaptive questions on the contents of stories require several capabilities such as semantic analysis, automated question generation and diagnosis of learners' answer sentences. This paper presents a method of selecting questions from a generated list to realize adaptive questions and answers. Our method filters out similar questions, and then selects questions by considering the difficulty, types and order. This paper also describes an evaluation of our method. As the result of our experiment demonstrates, we have found that our method generates a viable series of questions.

1. Introduction

It is common in second language learning, to answer questions on the contents of passages after listening and/or reading them. Such questions and answers (QA) in the target language is effective for acquiring practical skills because multiple language skills are required to answer the questions, in particular to grasp the contents of the story and the questions, as well as to compose answers. Many computer assisted language learning systems have been developed [1]. Some are equipped with exercise functions which ask about the contents of sentences. Most, however, use questions prepared beforehand [5, 6, 7]. Thus these have the problem that such systems will present questions without considering the learner's level of understanding because the number of prepared questions is limited.

In order to solve these problem, we are aiming to realize a QA function which provides adaptive questions on the surface semantics of English stories prepared by authors or learners. To realize the QA function, the following sub-functions are necessary: (1) to understand English sentences and to extract syntactic and semantic information, (2) to generate automatically various kinds of question sentences for presentation to learners who have varying degrees of comprehension, (3) to select adaptive questions from a set of generated question sentences, (4) to analyze learners' answer sentences and to diagnose errors and (5) to offer intelligent help for the correction of errors and the acquisition of correct knowledge by referring to the student models. In earlier studies, we have already implemented the sub-modules for the functions (1), (2), (4) and (5). This paper proposes an adaptive method of selecting questions for the function (3).

2. The outline of the QA function

Our QA function gives learners questions about the contents of a story. After studying the contents of the story by reading and/or listening, they answer the questions. Aims of our QA are both to train for conversation by using multiple skills through reading a story, listening to or reading questions and composing answers and to give learners a chance to realize their own state of understanding of, for example, vocabulary and grammar; and to practice usage through QA. To reduce the burden of memorizing the content of the story and to concentrate on composing sentences from memory, the length of any one passage in a single presentation set at about 5 or 6 sentences and QA on the surface meaning of the story is sufficient.

The QA function generates as many questions about the story as possible [3], and then, selects a suitable and purposive question.

3. Selecting Adaptive Questions

In order to achieve the aims mentioned in the previous chapter, the QA function needs to generate a series of questions according to the following principles instead of blindly giving questions.

(1) to give a tailored series of questions for each learner: questions that are too easy/difficult will reduce learner motivation. It is, therefore, desirable to give each learner questions of suitable difficulty.

(2) to make learners use as many skills as possible: The question generation module [3] generates four types of questions: a general question generated from one sentence, a special question generated from one sentence, a general question generated from more than one sentence and a special question generated from more than one sentence. Because these types of questions require different language skills on the part of the learner, it is necessary to give learners various types of questions.

(3) to give questions following the flow of the story: The QA function gives learners series of questions. It is desirable that each series covers the entire contents of a story instead of asking about the same part of the story. When the QA function gives such a series of questions, it is necessary to consider the order of questions because a series of questions without the consideration of the flow of a story will confuse learners.

(4) to avoid similar questions: Giving similar questions to already answered questions will reduce learner motivation. The QA function, therefore, needs to generate a series of questions after considering the history of QA.

The principles are classified into two groups: (4) is for avoiding undesirable questions and (1) - (3) are for selecting desirable questions. Our method, first, filters out undesirable questions in a series by using the following restrictions.

- Not selecting questions that have already been displayed.
- Selecting no more than two questions pertaining to a sentence when the learner's answer was correct.
- Not selecting questions with the same case and object as the previous question.

Next, our method tries to select desirable questions by referring to three factors: the difficulty of questions, the types of questions and the order of the questions. There is a best value for each selection factor and there is an acceptable range around the best value. It is necessary to select desirable questions by synthetically considering the three selection factors. Therefore, our method assigns the probability of selection to each question according to its desirability and selects one question at random. A way of assigning probability is as follows.

(a) the difficulty of questions: Our method tries to select questions with values of difficulty at 5 [1] points more/less than that of the previous question if a learner correctly/incorrectly answered the question. In order to realize such a selection, our method gives high probabilities to questions which have around the standard value of difficulty by referring to the normal distribution.

(b) the types of questions: In order to make learners use various skills, our method gives the probability of selection to each question type with the intention of avoiding the same question type as the previous one and selecting all types in a series.

(c) the order of questions: In order to select a series of questions to represent different sections of the story in an order same as that of the narration, our method defines the

[1] We have implemented a mechanism to calculate the difficulty of questions which reflects the learner's state of understanding [4]. In the previous study, we found that the threshold value used for the judgment of significant difference between two questions is 5.

areas of original sentences[2] according to a specified number of questions in a series and gives a probability of selection to each sentence in an area.

4. Evaluation

In order to confirm whether or not our method generates a good series, we compared a series of questions generated by our method with a series generated by a non-adaptive method. The non-adaptive method selects questions at random under the following three restrictions which can be implemented easily.

- to select questions following the flow of the story,
- not to select the same question twice, but
- to allow up to two questions from the same sentence in a story to be selected when a learner correctly answered such questions.

We used three stories from textbooks for Japanese junior high school students. Each story consists of six sentences. We set the number of questions for each series of questions to 4 and generated four series of questions about the contents of each story. We gave the stories and pairs of series generated by these two methods to subjects and asked them which they preferred. The number of subjects was 15. They were graduate and undergraduate students.

Each subject compared 12 pairs. Thus there were 180 pairs (12 pairs * 15 subjects) in total. The subjects judged that our method was superior in 119 pairs. Therefore, we have found that our method is significantly better by using the binomial test ($p<=0.01$).

5. Conclusions

This paper has presented the adaptive method of selecting questions by filtering out similar questions and considering the difficulty, type and order. As the result of the experiment shows, we have found that our method generates viable series of questions.

At present, the QA function has the grammar and lexical knowledge to analyze novice level English with the knowledge used in textbooks for Japanese junior high school students [2]. The function can interpret such sentences, generate questions automatically and realize adaptive QA for each learner. The levels of English ability are not limited, but we assume that learners have basic knowledge of English. Future work will concentrate on extending our method to longer stories.

References

[1] Gamper, J. & Knapp, J. (2001). A review of CALL Systems in Foreign Language Instruction", Proc. of the 10th International Conference on Artificial Intelligence in Education, 377-388.

[2] Kunichika, H., Takeuchi, A., & Otsuki, S. (1995). An Multimedia Language Learning Environment with Intelligent Tutor. In Chan, T., & Self, J. (Eds.). Emerging Computer Technologies in Education. VA: Association for the Advancement of Computing in Education. Ch.10.

[3] Kunichika, H., Katayama, T., Hirashima, T., & Takeuchi, A. (2001). Automated Question Generation Methods for Intelligent English Learning Systems and its Evaluation, Proc. of ICCE2001, 2, 1117-1124.

[4] Kunichika, H., Urushima, M., Hirashima, T., & Takeuchi, A. (2002). A Computational Method of Complexity of Questions on Contents of English Sentences and its Evaluation, Proc. of ICCE2002.

[5] Levin, L., & Evans, D. (1995). ALICE-chan: A Case Study in ICALL Theory and Practice. In Holland, V., Kaplan, J., & Sams, M. (Eds.). Intelligent Language Tutors: Theory Shaping Technology. NJ: Lawrence Erlbaum Associates Inc. Ch.5.

[6] Sawatpanit, M., Suthers, D. & Fleming, S. (2003). Evaluating a Second Language Learning Course Management System, Proc. of ICCE2003, 973-981.

[7] Villamañe, M. et al. (2001). Use and Evaluation of HEZINET; A System for Basque Language Learning, Proc.of ICCE2001.

[2] The question generation module generates questions by transforming one or more sentence(s) in a story. We call such sentences used for question generation "original sentences".

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

CM-DOM: a Concept Map Based Tool for Supervising Domain Acquisition

M. LARRAÑAGA, U. RUEDA, M. KEREJETA, J.A. ELORRIAGA, A. ARRUARTE
University of the Basque Country. P.K. 649, Donostia, E-20080

Abstract. In this paper CM-DOM, a concept map based graphical tool, is described. CM-DOM allows the supervision of the Domain Module acquisition from documents in a semi automatic way. This tool facilitates the inspection and refinement of the results of the acquired knowledge.

Introduction

Artificial Intelligence in Education can help in the document management for learning applications. In the information age, electronic documents constitute a valuable source of information that can be used in constructing the Domain Module of Computer Aided Instruction systems. Before incorporating documents in such systems a transformation process is required. In [1], a supervised method for acquiring the Domain Module from existing documents in Basque is presented. However, it is helpful to offer the user means for reviewing the results of the acquisition process. Therefore, an intuitive graphical tool will be useful in this task. Concept Maps have proved to be an appropriate means of representing and organising knowledge in a graphical way. They provide an intuitive and understandable description of the domain using graphic resources: *nodes* are used to represent the domain topics and *arcs* to express the relationships among them. CM-DOM is a concept map based tool that facilitates the inspection and refinement of the Domain Module.

In our approach, the Domain Module is composed of the Domain Ontology and a set of didactic resources formalised as Learning Objects [4]. The Domain Acquisition System performs a supervised three-phased process. (1) The domain ontology is built from the base document in two different steps. First the document table of contents is analysed to obtain an initial ontology. Next, this ontology is used to analyse the document body getting an enhanced ontology. (2) The domain ontology and general pedagogic knowledge are used to extract, classify and annote the Learning Objects [2]. Finally, (3) in the maintenance phase, new contents and Learning Objects are added by repeating the first two steps on new documents.

The approach chosen for the first phase combines Natural Language Processing with heuristic reasoning and has been tested with satisfying results [3]. However, due to the complexity of the information the human instructor has to deal with, the authors have found that a supervising tool is essential for checking the results of the acquisition process.

1. CM-DOM: a Concept Map Based DOMain Module Supervision Tool

CM-DOM is a graphic tool that aims to facilitate the inspection and refinement of the Domain Module. It is based on the generic concept map editor CM-ED [4] and profits from the advantages of the concept maps. This tool will be helpful to add/remove/change contents as well as pedagogical relationships or even creating user-defined SCORM compliant content sequencing organisations for particular learning situations [5]. CM-DOM will be useful not only to review the Domain Ontology but also to manage the whole Domain Module acquisition process. However, this paper focuses on the acquisition of the Domain Ontology.

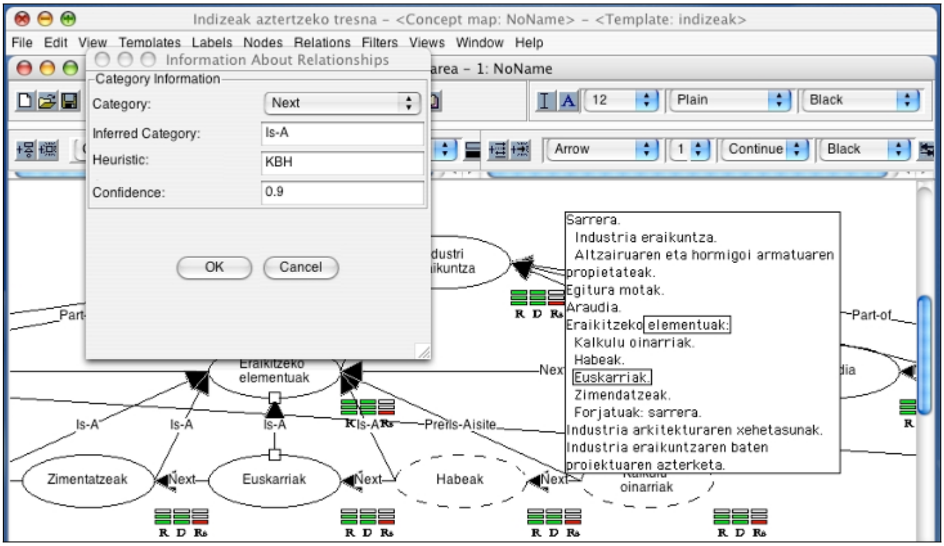
In the first phase, the Domain Ontology managed by CM-DOM is initially acquired by means of a heuristic process[3]. The ontology, which is stored in an XML file, contains the gathered topics and relationships together with information such as the relevance of each topic and the difficulty they entail for the learners. For each automatically gathered data the ontology also contains information about the heuristic that has been used to infer it and the confidence in the decision taken by the domain acquisition system. The instructional designer will use this information to acknowledge the system results or to make modifications.

CM-DOM provides the user with a graphical environment in which the tool represents the domain topics by means of nodes and the pedagogical relationships with arcs between nodes (see Figure 1). Meanwhile, the category of the topic is symbolised by the node shape (e.g. an oval node identifies a *concept* while a square node corresponds to a *procedure*), the category of the pedagogical relationship (*Is-A, Next,…*) is shown in the label of the arc. Besides the domain topics and the pedagogical relationships, other inferred information such as the topic relevance or the difficulty the topic may entail for the student is also presented. However, the information should be shown in a way that facilitates instructional designer's work and does not produce any additional cognitive overload. Therefore, the following graphical resources have been used. *Flags* are employed to visualise in a graphical way information about the topic relevance, difficulty and even the quantity of didactic resources associated with the particular topic. The *arc thickness* is used to express the strength of the relationship between topics, *i.e.* when two topics are frequently referenced together. Both, relationships and topics, are drawn with *dashed lines* until they have been checked by the user. Finally, the relationships and topics that have been inferred by low confidence heuristic decisions are drawn in a different configurable *colour* (*red* by default); this way the instructional designer knows which contents or relationships are more likely to be changed.

The information about a topic or a relationship is prompted to the human instructor when s/he double-clicks on it. The first time, the presented window shows the inferred category, the heuristics that have inferred it and the confidence in the decision. When the user double-clicks a domain topic, the window also shows information about its relevance, difficulty and quantity of related Learning Objects. The user can specify a different value for any of these characteristics by selecting it in the corresponding *ComboBox*. When the user closes the window clicking on the *OK* button, the concept map reflects the changes s/he has done either with different node shapes, relationship labels, thickness or flags. Once a relationship or a topic has been inspected, it is drawn with a black continuous line representing that it has been checked. In this way the user knows the stage of the supervision and the work still to be done. When working with the body of the document, the instructional designer can supervise the Learning Objects that have been linked to each domain topic and even remove or add to them.

During the reviewing process the instructional designer may change some information: topic or relationship category, topic relevance and so on. The operations performed by the human instructor modify the Domain Ontology. In addition, the level of confidence of the heuristics is updated according to the instructor's level of agreement with their results. In this way, the acquisition system implements an inductive learning process of the heuristics performance.

A usual concern when representing graphically the Domain Module is the scalability of the approach. In order to face this important issue the tool implements two mechanisms. On the one hand, the user can contract and expand parts of the Concept Map. On the other hand, the tool provides a *view* mechanism that allows the user to work with parts of the Domain Ontology. For example, setting the corresponding view the user may supervise only one kind of relationship.



Indizeak aztertzeko tresna – <Concept map: NoName> – <Template: indizeak>

Information About Relationships

Category Information

Category: Next

Inferred Category: Is-A

Heuristic: KBH

Confidence: 0.9

Sarrera.
Industria eraikuntza.
Altzairuaren eta hormigoi armatuaren propietateak.
Egitura motak.
Araudia.
Eraikitzeko elementuak:
Kalkulu oinarriak.
Habeak.
Euskarriak.
Zimendatzeak.
Forjatuak: sarrera.
Industria arkitekturaren xehetasunak.
Industria eraikuntzaren baten proiektuaren azterketa.

Figure 1: CM-DOM snapshot

Figure 1 illustrates how the instructional designer supervises a pedagogical relationship that has been inferred by the KBH heuristic, which uses a set of keywords (i.e. *elementuak* which means *elements* in Basque) to detect the *Is-A* relationship.

2. Conclusions

In this paper, CM-DOM a concept map based tool is presented. CM-DOM is a graphical tool developed with the aim of lightening the work of the instructional designer during the supervision of the domain acquisition process for Technology Supported Learning System. The tool uses graphical resources such as line styles, colours, node shapes, labels and flags to present the information as simply and intuitively as possible. The tool provides mechanisms to work with parts of the Domain Ontology in order to face scalability. CM-DOM uses the concept map of the domain as the backbone to manage the whole life-cycle of the Domain Module acquisition process, including the ontology building, the definition of Learning Objects and even the maintenance phase. The next step in the domain acquisition process is the semiautomatic identification and extraction of Learning Objects from documents.

Acknowledgements: This work is funded by the University of the Basque Country (UPV00141.226-T-15948/2004), the CICYT (TIC2002-03141) and the Gipuzkoa Council in an European Union program.

References

1. Larrañaga, M. *Enhancing ITS building process with semi-automatic domain acquisition using ontologies and NLP techniques.* in *Young Researches track of the Intelligent Tutoring Systems (ITS 2002)*. 2002. Biarritz.
2. LTSC, *IEEE P1484.12*. 2001.
3. Larrañaga, M., Rueda, U., Elorriaga, J.A., and Arruarte, A. *Acquisition of the Domain Structure from Document Indexes Using Heuristic Reasoning.* in *7th International Conference on Intelligent Tutoring Systems*. 2004. Maceió, Alagoas, Brazil: Springer-Verlag.
4. Arruarte, A., Elorriaga, J.A., and Rueda, U. *A Template Based Concept Mapping Tool for Computer-Aided Learning.* in *IEEE International Conference on Advanced Learning Technologies*. 2001: IEEE Computer Society.
5. Advanced Distributed Learning, *SCORM Content Aggregation Model version 1.3.1*. 2004.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Using FAQ as a Case Base for Intelligent Tutoring

Demetrius Ribeiro LIMA[1], Marta Costa ROSATELLI[2]
[1]*Curso de Pós-Graduação Ciência da Computação, Universidade Federal Santa Catarina*
Campus Universitário, Trindade, Florianópolis – SC, 88040-900, Brazil
[2]*Programa de Mestrado em Informática, Universidade Católica de Santos*
R. Dr. Carvalho de Mendonça, 144, Vila Mathias, Santos, SP, 11070-906, Brazil

Abstract. This paper presents an Intelligent Tutoring System that was designed and integrated to a web-based distance learning environment. In this system, the Frequently Asked Questions is the knowledge base of a Case-Based Reasoning module that is used to retrieve situations similar to the ones currently presented by the student, i.e., to search for answers to the students' queries.

1. Introduction

In distance learning environments, the queries that result from the process of interaction between the student and the course contents allow constructing a knowledge base in which questions (specially the frequently asked ones), and respective answers, can be organized, retrieved, adapted, and reused in similar situations. In this sense, Frequently Asked Questions (FAQ) can support distant learning as the main idea behind this kind of tool is to register the consensus of opinions and issues of several students in a single question. As the human tutor responds to this question, the answer can be useful to other students. Besides, the FAQ tool that is usually included in distance learning environments has a structure that is in conformance with the requisites of the Case Based Reasoning (CBR) stages [2].

This paper presents an Intelligent Tutoring System (ITS) designed and integrated to a web-based distance learning environment. It aims to help the human tutor in accompanying and providing individualised support to the distant student along the learning process. This is accomplishing by a CBR module in which the FAQ is the knowledge base, and that retrieves situations (cases) similar to the ones currently presented by the student.

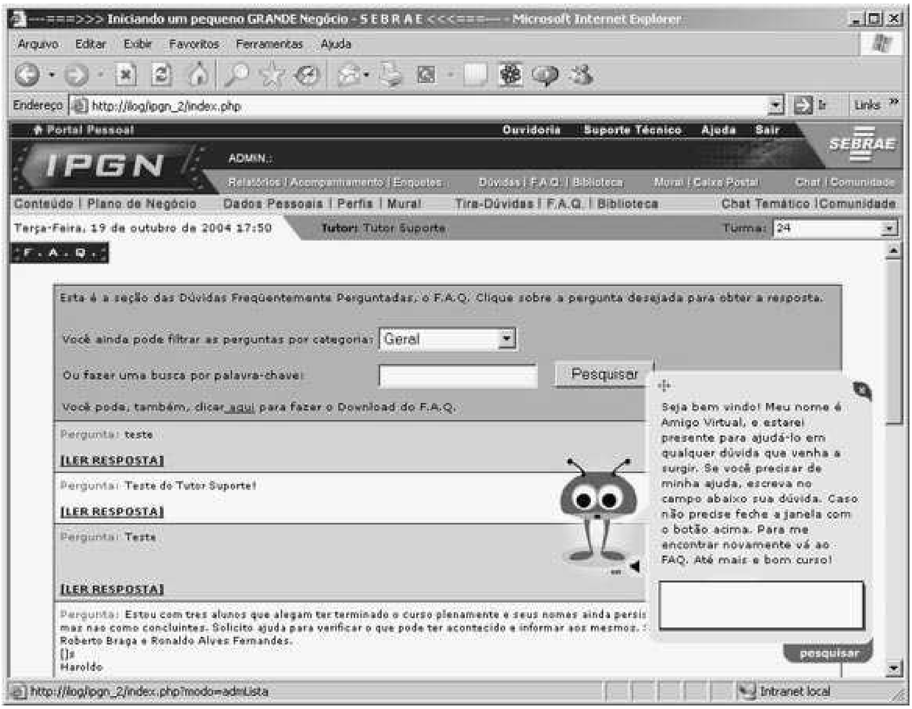
2. Web-based Learning Environment

The web-based learning environment (see Fig. 1) includes functionalities and tools for two main users: student and tutor. A distance course on Entrepreneurship has been offered since 2001 to the general public through this learning environment. The course, which aims to train students in the planning aspects of starting a new business, is free and is part of the training program of an organization that supports small and medium enterprises. Usually, 350 classes of 200 students attend this course every year and around 250,000 people were already trained by this course.

3. System Overview

The system architecture depicted in Fig. 2 shows the FAQ as the case base of the ITS. The student model takes into consideration both the student use and usage data. The Nearest

Neighbour [4] is the classification method and the Manhattan distance is used to calculate the similarity between the student current state and one of the profiles defined - novice, intermediate, and advanced. Based on the student current state, the tutor model (1) initiates a system intervention if it identifies the need to provide support to the student, and (2) if appropriate, retrieves a case from the case base.


---===>>> Iniciando um pequeno GRANDE Negócio - S E B R A E <<<===--- - Microsoft Internet Explorer
Arquivo Editar Exibir Favoritos Ferramentas Ajuda
Endereço http://ilog/ipgn_2/index.php
Ir
Links »
Portal Pessoal
Ouvidoria Suporte Técnico Ajuda Sair
SEBRAE
IPGN
ADMIN.:
Relatórios | Acompanhamento | Enquetes
Dúvidas | F.A.Q. | Biblioteca
Mural | Caixa Postal
Chat | Comunidade
Conteúdo | Plano de Negócio
Dados Pessoais | Perfis | Mural
Tira-Dúvidas | F.A.Q. | Biblioteca
Chat Temático |Comunidade
Terça-Feira, 19 de outubro de 2004 17:50
Tutor: Tutor Suporte
Turma: 24
F . A . Q .
Pesquisar
[LER RESPOSTA]
[LER RESPOSTA]
[LER RESPOSTA]
pesquisar
http://ilog/ipgn_2/index.php?modo=admLista
Intranet local

Esta é a seção das Dúvidas Freqüentemente Perguntadas, o F.A.Q. Clique sobre a pergunta desejada para obter a resposta.

Você ainda pode filtrar as perguntas por categoria: Geral

Ou fazer uma busca por palavra-chave:

Você pode, também, clicar aqui para fazer o Download do F.A.Q.

Pergunta: teste

Pergunta: Teste do Tutor Suporte!

Pergunta: Teste

Pergunta: Estou com tres alunos que alegam ter terminado o curso plenamente e seus nomes ainda persis
mas nao como concluintes. Solicito ajuda para verificar o que pode ter acontecido e informar aos mesmos. :
Roberto Braga e Ronaldo Alves Fernandes.
[]s
Haroldo

Seja bem vindo! Meu nome é Amigo Virtual, e estarei presente para ajudá-lo em qualquer dúvida que venha a surgir. Se você precisar de minha ajuda, escreva no campo abaixo sua dúvida. Caso não precise feche a janela com o botão acima. Para me encontrar novamente vá ao FAQ. Até mais e bom curso!

Figure 1. Web-based learning environment

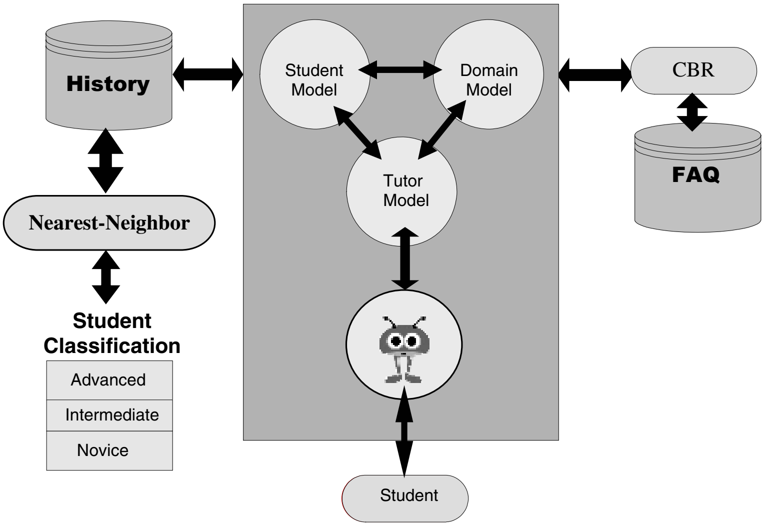


Figure 2. System architecture

4 FAQ as the Case Base

The FAQ includes not only the frequently asked questions, but also the questions that the human tutors judge to be of interest to all students. The Q-A (Question-Answer) pairs of the FAQ compose the case base that is used to retrieve situations similar to the ones currently presented by the student. The Q-A pair is the Case representation that is stored in the case base and Question is the index of the knowledge contained in the Answer.

As case attributes are strings, the TFIDF (Term Frequency Inverse Document Frequency) method, which represents document as a weights vector and each weight as a document term (vector model), is used to retrieve similar cases. The similarity between two documents is measured by the cosine of the angle between the vectors that correspond to these documents [5]. The vector model is defined by indicating non binary weights to the terms that are the indexes (Query and Cases) and that are used to compute the degree of similarity between each stored document and the student query. Thus, the system retrieves the documents (Cases) classifying and presenting them in a decreasing order of degree of similarity in relation to the query presented. The text entered by the student through the ITS interface is the Query. The terms considered as indexes are the ones that are not included in the Stop List (i.e., a list that contains all the terms that are not taken into account in the comparison).

The weights of the terms that are indexes are related with clustering the group of characteristics that provide the intra-cluster classification and the group of characteristics that provide the quantification of inter-clusters dissimilarity. In the vector model, the intra-cluster similarity is measured by the frequency of the term inside the Case (term frequency) [3]. The inter-cluster dissimilarity is quantified by the inverse of the document frequency (inverse document frequency) in the Case collection, as terms that appear in many cases are not useful to distinguish the relevant from the non relevant ones [1].

The CBR module uses both derivational and structural adaptation [4]. When the ITS interacts with the student, he or she has different options of actions that provides the data for the system adaptation. The representation of the FAQ in the case base includes the attributes: Question, Answer, InclusionDate, and ID. An additional table - Feedback - is related to the FAQ includes the attributes: Question, Answer, ID, IDFaq, and Status.

5. Conclusion and Further Work

This paper presented an ITS that uses the FAQ as a case base of a CBR module to provide support to students in distance learning. Case retrieval is carried out using the TFIDF method and vector model. Derivational and structural adaptation are used for adding new cases to the case base and updating the Case attributes. Further work includes categorizing Q-A pairs in the case base and using the categories as indexes to improve Case retrieval.

References

[1] Baeza-Yates, R. & Ribeiro-Neto, B. (1999). Modern Information Retrieval. New York: ACM Press.

[2] Burke, R. D., Hammond, K. J., Kulyukin, V. A., Lytinen, S. L., Tomuro, N. & Schoenberg, S. (1997). Question Answering from Frequently-Asked Question Files: Experiences with the FAQ Finder System. Technical Report TR-97-05, University of Chicago.

[3] Salton, G. & Buckley, C. (1987). Term Weighting Approaches in Automatic Text Retrieval. Technical Report: TR87-881, Department of Computer Science, Cornell University.

[4] Watson, I. (1997). Applying Case-Based Reasoning: Techniques for Enterprise Systems. San Francisco: Morgan Kaufmann.

[5] Zukerman, I. & Albrecht, D. W. (2001). Predictive statistical models for user modeling. User Modeling and User-Adapted Interaction 11(1-4), 5-18.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Alignment-Based Tools for Translation Self-Learning[1]

J. Gabriel Pereira Lopes and

Tiago Ildefonso

CITI/DI/FCT/Universidade Nova de Lisboa, Portugal
{gpl | tiago.ildefonso}@di.fct.unl.pt

Marcelo S. Pimenta

Instituto de Informática/UFRGS, Brazil
mpimenta@inf.ufrgs.br

– **Abstract**: Effective use of translation products and services depends on professional and learner translators familiarity with different translation procedures, strategies and tools, currently available for solving specific translation problems. In this poster we will focus on the use of language independent alignment-based technology, applied to parallel corpora, for effectively aiding translation learning and translation quality improvement, both at-work and at-the-school.

1.Introduction

Translation has undergone considerable changes over the last decade. Translation technology has gained wider acceptance and is currently used by professional translators. Computer-aided translation (CAT) tools entered translation services due to their contribution for improving translators productivity and translation quality. Information Retrieval technology has also affected translation research, as witnessed by the rising interest in corpus-based approaches to translation studies. Such changes have brought new opportunities and new challenges both for translation and for translators training. In this paper, we will show that CAT tools, that have already proved to be effective in providing linguistic help in translation situations, can also effectively support translation self-instruction at-the-work and at-the-school.

Despite the considerable scientific production on CAT, surprisingly, there is little research work devoted to translation learning and to the use of CAT tools in the classroom. But there are exceptions. Somers (2001) discusses strategies for Machine Translation (MT) teaching. Balkan et al. (1997) surveyed tools and techniques for MT teaching. Kenny and Way (2001) report on experiences in MT and translation technologies teaching. Fictumová (2004) discusses the use of the open source e-learning environment LMS Moodle for translators and interpreters. All of them agree on the need to bring translation instruction closer to the real world of professional translation in order to cope with imminent changes and challenges in the translation market. The same applies to the need to learn how to use translation tools (Somers, 2003; Forcada, 2000). However, material for self-learning, for supporting autodidactic translation students and for training professional translators, is rare.

In this poster we will focus on autonomous self-learning of translation, both at-work and at-the-school. For this purpose we will use parallel text alignment-based tools produced in the framework of TRADAUT-PT, a MLIS European project, for supporting lexicon coders to speed up the construction of lexical entries for the various kinds of lexicons used by Systran MT systems. Those tools include a language independent aligner (Ribeiro et al, 2000; Ildefonso et al, 2005), a bilingual concordancer for each pair of languages considered, monolingual concordancers, an extractor of single word and multi-word translations, a web interface for enabling different clients to work, safely and independently, at-the-distance, using these tools and MT systems. As our aligner is language independent and partitions

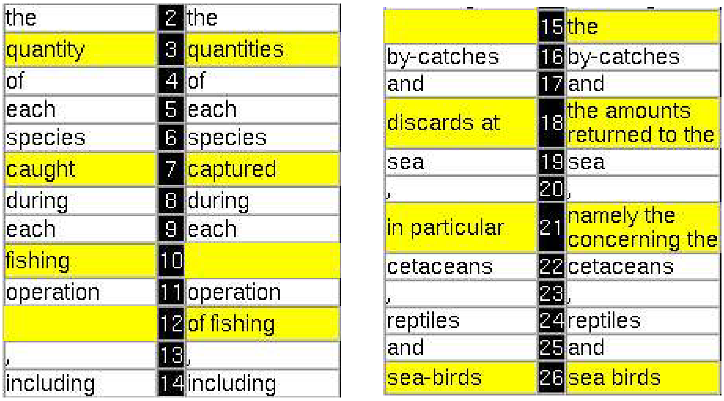
[1] This work has been granted by CAPES/GRICES Brazilian-Portuguese joint project We-Learn.

input parallel texts into parallel segments (text stretches that are translations of each other or a common source text stretch) finer grained than the sentence, it can also align human made and golden standard human translations (figure 1).

2. Using our alignment-based set of tools for translation instruction

In this poster, we will focus on an specific use of our aligner for **translation quality evaluation** and on the use of our **bilingual concordancer** for helping the learner to check why her/his translation failed to achieve golden standard quality**.**

In general, the closer a translation and its golden standard translation are, the better that translation is. As a given sentence may be translated in many ways, a student need to know why his/her own translation diverges from a golden standard. Let us see how this can be achieved using our tools. Consider the excerpt of the translation made by Systran MT system, from Portuguese into English, for the European Council Regulation (EC) No 1239/98 stretch, represented in figure 1, at the right side of the two screen dumps captured. At the left side of those screen dumps it is depicted the aligned golden standard.



the	2	the
quantity	3	quantities
of	4	of
each	5	each
species	6	species
caught	7	captured
during	8	during
each	9	each
fishing	10	
operation	11	operation
	12	of fishing
,	13	,
including	14	including

	15	the
by-catches	16	by-catches
and	17	and
discards at	18	the amounts returned to the
sea	19	sea
,	20	,
in particular	21	namely the concerning the
cetaceans	22	cetaceans
,	23	,
reptiles	24	reptiles
and	25	and
sea-birds	26	sea birds

Figure1: Alignment of golden standard in English, at the left column, with corresponding translations by Systran MT system, at the right column.

Used Golden standard is: "*the quantity of each species caught during each fishing operation, including by-catches and discards at sea, in particular cetaceans, reptiles and sea-birds,"*. Corresponding Portuguese source text is: "as quantidades de cada espécie capturadas durante cada operação de pesca, incluindo as capturas acessórias e as quantidades devolvidas ao mar, nomeadamente as relativas aos cetáceos, répteis e aves marinhas,".

In figure 1, first difference is the use of "quantities" instead of "quantity", for translating "quantidades". Consulting our bilingual concordancer for "as quantidades de cada espécie" we get 10 times "the quantity of each species", 7 times 'th e quantities of each species", and twice "the amount of each species". In this case, we may consider proposed translation as correct. Second difference, related to the use of "captured" instead of "caught", shows that "harvested" occurs 5 times, "caught" 3 times, "taken" twice, "fished" once, apart from 4 other singular occurrences. Regarding the preference for "fishing operation", by consulting our bilingual concordancer, we observe that "fishing operation" occurs 8 times and "haul" 7 times. Such observation should lead a student to reflect and, if necessary, to interact with her/his teacher. Fourth difference relates to the use of plural definite article in English. Fifth difference relates to the literal translation of "quantidades devolvidas ao mar" and the preferred translation "discards at the sea". Sixth difference relates to the literal translation "as relativas", not taking into account that, in this case, "as" is a demonstrative pronoun and so, it might be translated as "those", giving rise to "namely those concerning cetaceans". This translation, though different, would be perfectly acceptable. Again plural definite article should be deleted. Last difference, would lead the student to learn, by using the concordancer, that "seabirds" occur 5 times and "sea-birds" twice. Despite these numbers, my spell checker proposes "sea birds" for correcting "seabirds".

3.Conclusions

We assume that students have a certain competence in the non-native languages they are working on. So, in this paper we focussed on issues related to hands-on translation training. The alignment-based tools we propose help the students to understand basic translation notions (equivalence or lack of it, translatability, loss, compensation, faithfulness, naturalness, form, content, meaning etc.). They also help students to decide, choose and evaluate their own performance during a translation process. Proposed environment additionally introduces students to existing translation technology. Students will draw their conclusions from their practical experience, by finding and solving translation problems and becoming familiar with different translation procedures and strategies available to solve those problems.

Our language independent approach to parallel text alignment gives rise to parallel text segments with a length shorter than the sentence. Moreover, it may be directly applied to the alignment of any translation produced either by a student or by a machine and align it with a golden standard (a parallel text produced by a professional translator). This way, most part of parallel segments will have a length equal to one word. Differences generally occur in larger segments. A traditional aligner, working at the sentence level, would be unable to produce such a low grained alignment, thus making more difficult the visualization of differences. By signalizing differently parallel segments having different content, a student may easily view the differences between a golden standard and his/her own translation.

References

L. Balkan et al. 1997. Tools and Techniques for MT Teaching. Survey Report, Essex University. http://clwww.essex.ac.uk/group/projects/MTforTeaching/index.html

M Carl and A. Way (2003). Recent Advances in Example-Based Machine Translation. Kuwer Academic Publishers. Dordrecht: Netherlands.

J. Fictumová et al. E-Learning for Translators and Interpreters – the case of LMS Moodle. IN:Consortium for Training Translation Teachers (CTTT), Project Papers, 2004.

M. L. Forcada, "Learning machine translation strategies using commercial systems: discovering word-reordering rules", in *Proceedings MT 2000: Machine Translation and Multilingual Applications in the New Millennium* (Exeter, UK, November 2000).

T. Ildefonso and G. P. Lopes (2005) "Longest Sorted Sequence for parallel text alignment". Proceedings of Eurocast'05. (to be published by Springer).

D. Kenny, A. Way. Teaching Machine Translation & Translation Technology: A Contrastive Study. In: Proceedings of Teaching Machine Translation Workshop at VIII MT Summit, Santiago de Compostela, Spain, 2001.

A. Ribeiro, J.G. P. Lopes and J. Mexia 2000. "A self Learning Method of Parallel texts alignment". In John White (ed,) Envisioning Machine Translation in the Information Future. 4th AMTA 2000, Cuernavaca, Mexico, Proceedings. LNAI series 1934,

H.Somers. *2001.* "Three Perspectives on MT in the Classroom". IN: Teaching Machine Translation Workshop at MT Summit VIII, Santiago de Compostela, p. 25-29

Somers, H. (2003): «The translator' sworkstation», In: Somers, H. (ed), Computers and Translation. Amsterdam/Filadelfia: John Benjamins, 13-31.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Implementing Analogies Using APE Rules in an Electronic Tutoring System

Evelyn LULIS[1], Reva FREEDMAN[2], and Martha EVENS[3]
elulis@cti.depaul.edu, freedman@cs.niu.edu, evens@iit.edu
[1]*CTI, DePaul University, Chicago, IL, USA*
[2]*Dept. of Computer Science, Northern Illinois University, DeKalb, IL, USA*
[3]*Dept. of Computer Science, Illinois Institute of Technology, Chicago, IL, USA*

Abstract. In a corpus of eighty-one human tutoring sessions in physiology, the use of analogy to help students make correct inferences was successful 81% of the time. As a result, we are adding the most important tutor-proposed analogies that have a base in the domain to our intelligent tutoring system CIRCSIM-Tutor. We are using the APE dialogue planner because it allows for both hierarchical schemas and changes to the plan when the tutor wants to correct a student's error.

1. Introduction: Analogies in CIRCSIM-Tutor

CIRCSIM-Tutor [1, 2] is a dialogue-based intelligent tutoring system used at Rush Medical College to tutor first year medical students in blood pressure control by the autonomic nervous system. A corpus of eighty-one hour-long sessions conducted by two physiology professors at Rush has been used as a basis for developing the strategies and language used in the tutor. Analysis of the corpus [3] yielded fifty-one analogies proposed by the tutors and eight proposed by the students. In the thirty-seven cases where the tutor requested an inference after the use of analogy, the student made correct inferences thirty times (81% success rate). With a one-hour session with the existing version of the tutor, students performed better on the post-test than the pre-test ($p<.001$) [1]. The addition of analogical reasoning to the tutor will further help students understand blood pressure control.

We have identified [4, 5] several analogies as targets for implementation, including another neural variable, the reservoir model [6, 7], the compliant structure model [6, 7]), the pressure/flow/resistance model [6, 7], and the accelerator and brake model. We are currently implementing some of them using the APE dialogue planner [8]. This paper discusses the dialogue schemas used by the implementation and the related APE plans. We show how features of APE, in particular the ability to make changes to plans in progress in response to student errors, are useful in the implementation of analogies.

2. Implementation

The most frequently used pattern, *another-neural-variable,* was chosen for our original implementation. This analogy makes use of the fact that all three neurally controlled variables behave in the same manner. After tutoring one of them, the tutor prompts the student to make an analogy between it and one or more of the others:

K3-tu-65-4: Are there no other neurally controlled variables that
would change at the same time?

K3-st-66-1: CC?
K3-tu-67-1: How would it change?
K3-st-68-1: Parasympathetic reflex would decrease CC.

Like most task-oriented dialogues, CIRCSIM-Tutor dialogues have a hierarchical structure [9]. In addition, obtaining the best student performance requires providing specific responses to distinct student errors, which implies changes to the hierarchical structure as the system is running. APE [8], a dialogue management system based on reactive planning, provides a flexible and robust platform that can handle both of these requirements.

The *another-neural-variable* pattern can be summarized as follows:

```
If the next variable to tutor is neural
and its value can change (i.e., it is not clamped)
and a neural variable has already been tutored
then the tutor proposes an analogy to the student by asking, e.g.,
    "Can you think of another variable that is neurally controlled?"
    If the student answers correctly
        then the tutor requests an inference from one of those variables, by
            asking, e.g.: "What happens to <variable> in that case?"
        If the student answers correctly
            the implicit version of the analogy was successful
              /* so the tutor moves on to the next topic */
        else
              /* the explicit version of the analogy is required */
              the tutor asks (or tells) the student how to map the analogs
              the tutor asks (or tells) the student how to map the relationships
              the tutor prompts the student to make an inference to
                determine understanding.
```

If the original *if* statement is not satisfied, we want to provide a specific correction for each type of student error in order to remediate the defects in the student's understanding. For example, if the student chooses an incorrect base variable for the analogy, the tutor can ask the student to reread the procedure description, remind the student of a basic fact, or push a new goal on the agenda [5].

Below we show three APE plans. The first plan gives the conditions under which the *another-neural-variable* (ANV) plan applies and determines the target of the analogy, using as the base variable a neural variable that has already been corrected:

```
(def-operator T-tutors-ANV
   :goal (did-neural-variable)
   :precond ((is-current-problem ?pb)
             (is-current-variable ?target-vbl)
             (is-neural ?target-vbl)
             (not (is-clamped ?pb ?target-vbl))
             (is-neural ?base-vbl)
             (is-corrected ?base-vbl))
   :recipe  ((goal (get-analogous-vbl ?base-vbl ?target-vbl))
             (retract (no-error (tried-ANV-short)))  ;;; initialize flag
             (goal (did-ANV-analogy ?base-vbl ?target-vbl)))
```

The second and third plans show the short and long forms of the analogy, respectively. A flag is used to ensure that the short form is tried first. In that version, the tutor just sees whether the student can correctly make the inference about the value of the variable. In the long form, tutor and student explicitly map the relationships.

```
(def-operator T-tutors-ANV-short
  :goal (did-ANV-analogy ?base-vbl ?target-vbl)
  :precond ((not (tried-ANV-short)))                ;;; flag is false
  :recipe  ((assert (tried-ANV-short))              ;;; set flag to true
            (goal (get-value ?target-vbl)))

(def-operator T-tutors-ANV-full
  :goal (did-ANV-analogy ?base-vbl ?target-vbl)
  :precond ((tried-ANV-short))                      ;;; flag is true
  :recipe  ((goal (did-map-analogs ...))
            (goal (did-map-relationships ...))
            (goal (get-value ?target-vbl))
```

3. Conclusion

We are adding tutoring by analogy to CIRCSIM-Tutor, a dialogue-based intelligent tutoring system in physiology. In a corpus of eighty-one human tutoring sessions, the use of analogy was successful 81% of the time. As a result, we are implementing the tutor-proposed analogies with a basis in the domain using the APE dialogue planner. In this paper we give examples of our dialogue schemas and the implementation of the schemas as APE plans. In addition to being fast and robust, APE allows for both hierarchical schemas and changes to the hierarchy required to provide differential feedback to students depending on their errors.

Acknowledgments

This work was partially supported by the Cognitive Science Program, Office of Naval Research, Grant 00014-00-1-0660 (Stanford University) and Grants No. N00014-94-1-0338 and N00014-02-1-0442 (Illinois Institute of Technology). The content does not reflect the position or policy of the government and no official endorsement should be inferred.

References

[1] Michael, J., Rovick, A., Glass, M., Zhou, Y. and Evens, M. (2003). Learning from a Computer Tutor with Natural Language Capabilities. *Interactive Learning Environments,* 11(3): 233–262.

[2] Evens, M. and Michael, J. (in press). *One-on-one tutoring by humans and machines.* Mahwah, NJ: Erlbaum.

[3] Lulis, E., Evens, M. and Michael, J. (2003). Representation of analogies found in human tutoring sessions. In Proceedings of the Second IASTED International Conference on Information and Knowledge Sharing, Scottsdale. Anaheim, CA: ACTA Press. Pp. 88–93.

[4] Lulis, E., Michael, J. and Evens, M. (2004a). Using qualitative reasoning in the classroom and in electronic teaching systems. In Papers from the Workshop on Qualitative Reasoning, Northwestern University, Evanston, IL. Pp. 121–127.

[5] Lulis, E., Michael, J. and Evens, M. (2004b). Implementing analogies in an electronic tutoring system. In Proceedings of ITS 2004. Berlin: Springer. LNCS 3220. Pp. 751–761.

[6] Modell, H. (2000). How to help students understand physiology? Emphasize general models. *Advances in Physiology Education*, 23: 101–107.

[7] Michael, J. and Modell, H. (2003). *Active learning in the college and secondary science classroom: A model for helping the learner to learn.* Mahwah, NJ: Erlbaum.

[8] Freedman, R. (2000). Using a Reactive Planner as the Basis for a Dialogue Agent. In *Proceedings of FLAIRS 2000,* Orlando. Pp. 203–208.

[9] Grosz, B. (1977). The Representation and Use of Focus in a System for Understanding Dialogs. In *Proceedings of IJCAI '77,* Cambridge, MA. Pp. 67–76.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Interoperability Issues in Authoring Interactive Activities

Manolis Mavrikis[1] and Charles Hunn
School of Mathematics
The University of Edinburgh
M.Mavrikis@ed.ac.uk, C.Hunn@ed.ac.uk

This paper describes an attempt to integrate the suite of Cognitive Tutors Authoring Tools (CTAT) with an ITS for exploratory activities (DANTE). Despite their differences we provide positive indicators for integration and solutions that could be of use to developers of similar frameworks or authoring tools. In addition, by translating CTAT's XML representation of procedural activities to a more general one, used by a web-based ILE (WaLLiS), we identify missing information that need to be encoded after the translation and provide requirements for future changes or development of other authoring tools.

1. Introduction

It is well known that the development of ITS and especially their content and exercises is a very expensive, complex and time consuming process [1,2,3]. For projects with limited resources this limits the potential of the system, the depth of domain coverage, and consequently its impact [2,3]. Developing an authoring 'shell' consisting of reusable interface and tutoring components is an effective approach but only reduces the system's complexity, it does not overcome the major challenge of enabling domain experts to be directly involved in authoring [2]. This can only be achieved by appropriate authoring tools which benefit in that they have the potential to decrease the time, cost and skill threshold of development, while enabling rapid prototyping [4]. Additionally, the expense of system development can be reduced by designing with interoperability and component reuse in mind. This approach has been described in [5] and allows developers to integrate third party components rather than having to develop them from scratch. On the other hand, it is often the case that semantic interoperability is not always achievable. Additional information, implicitly encoded in one system, needs to be explicitly annotated in other systems and too specific information is often difficult to generalise.

Our research involves two environments: DANTE [6] and WaLLiS [7] and a suite of Cognitive Tutor Authoring Tools - CTAT [3,4]. In an attempt to integrate the latter and use it for authoring exercises we describe interoperability issues, and solutions to facilitate the integration and raise awareness for other similar systems.

2. Dynamic Authoring aNd Tutoring Environment (DANTE)

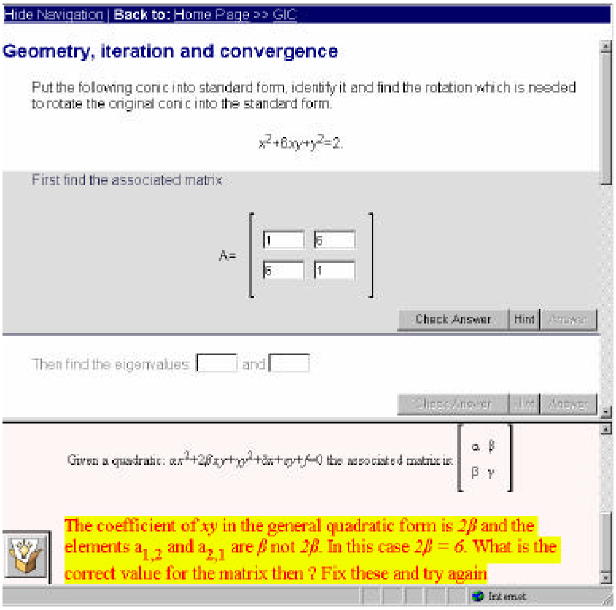
DANTE is an intelligent environment that was designed to overcome limitations of computer based environments which foster microworld-like or exploratory and self-learning interactive activities. The main rational behind this was that, despite the success of microworlds there are many cases, especially with self-learning material, during which students are based solely on their own perception and understanding of the concept. One of the main considerations of DANTE was to have an easy (script-like) way for authoring activities. The details of its implementation are described elsewhere [6,7,8]. DANTE is integrated in a web-based system (WaLLiS), used in the School of Mathematics of the University of Edinburgh [7], to deliver its exploratory activities.

[1] Correspondance: JCMB, School of Mathematics, University of Edinburgh, Mayfield Road, EH93JZ, UK.

We described previously attempts to integrate CTAT and DANTE [see 8, 9] and we have now identified further interoperability issues with the representation of exploratory activities that DANTE provides in the CTAT framework. For example, one of the activities asks the student to explore possible conic sections and identify them. For such activities, which involve experimentation with some graphical or other representational component in the interface, the tutor agent should be able to monitor the continuously changing values of the interface component in order to monitor the students' progress. This is not possible in CTAT. In addition, once a student has identified a meaningful configuration of a representational object in the interface, the correct input may be somewhere within a range of correct values rather than a specific one that CTAT expects. To address this problem, we modified our component in an ad-hoc way. However, this moves the goal representation to the component itself and hinders its reusability. Appropriate constructs should be defined in the authoring tool to allow the author to specify ranges of actions that have the same effect.

5. Employing CTAT as an authoring tool for web-based interactive activities

The aforementioned limitations of CTAT are related to the fact that it was not designed for exploratory activities per se. Therefore, we decided to test it as an authoring tool for the procedural activities that WaLLiS employs. In this case, the system's intelligence and feedback relies on the choices the author makes in an XML file which represents a comprehensive graph of the exercise that automatically gets translated to JavaScript (see figure 3). Despite their limitations these activities have proven to be effective and provide a less expensive solution for procedural activities. However, particular steps and alternative paths of the solution tree have to be authored explicitly; still a time consuming process.



Geometry, iteration and convergence

Put the following conic into standard form, identify it and find the rotation which is needed to rotate the original conic into the standard form.

$x^2+6xy+y^2=2.$

First find the associated matrix

A= [1, 6; 6, 1]

Then find the eigenvalues and

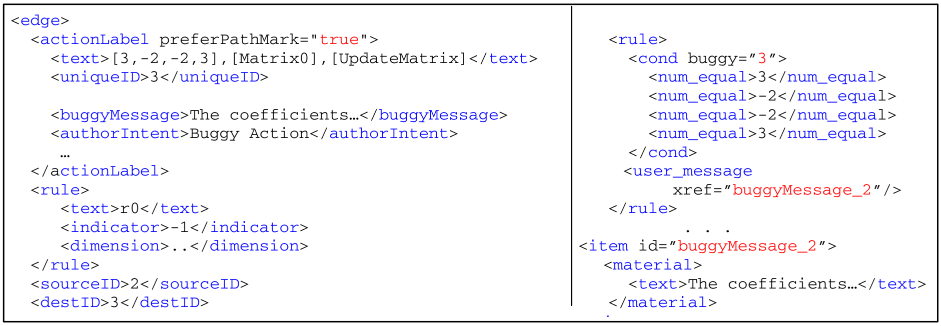
Given a quadratic: $\alpha x^2+2\beta xy+\gamma y^2+\delta x+\epsilon y+f=0$ the associated matrix is $\begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix}$

The coefficient of *xy* in the general quadratic form is *2β* and the elements $a_{1,2}$ and $a_{2,1}$ are *β* not *2β*. In this case *2β = 6*. What is the correct value for the matrix then ? Fix these and try again

Figure 3. Exercises in WaLLiS have an XML representation that allows authoring multiple parts, alternative solution paths, and adaptive feedback based on common misconceptions (see [10] for more details).
They can be authored in a general way based on randomised constants and mathematical expressions. This provides a flexibility and saves significant time. For example, the exercise on the left is produced by randomly generating two values that represent the eignevalues and form the quadratic equation in question. Then, the rest of the exercise and it feedback can be authored based on these values.
However, the complicated representation, the graph structure and the links between states makes it hard to validate and debug without an authoring tool.

We exploited CTAT's facility to build custom controls and a dormin widget [3,4] to represent activities involving matrices thus developing a pseudo-tutor that can teach conversion of quadratic equations to their standard form [8]. By using the pseudo-tutor we have the model in a format that can be tested and viewed by domain experts, reducing the model-building and debugging phase. We use the resulting tree represented in CTAT's XML format and transform its relevant parts to the WaLLiS format. However, WaLLiS' student model uses information from the exercise to be able to guide the learner further. For instance, common mistakes are used to provide further exercises that possibly address the learner's misconception. As one might expect, the missing information after the translation from CTAT is crucial. The way an author adds knowledge labels and buggy rules in CTAT is arbitrary, and hence not interoperable. In the future we have to make sure that the language used to describe these is appropriate and can be mapped to other systems. For example, for WaLLiS an internal ontology (for instance, specified in OMDoc [11]) has the potential to fully represent the domain. With the definition of an appropriate domain and with a GUI component an author would be able to choose concepts that belong in a controlled language rather than her arbitrary choice.

A limitation of the approach is that the produced XML tree is specific to the particular values the author chose. As [4] describes, demonstrating alternative paths can be really tedious. Fortunately, some of the exercises can be generated by randomising variables of appropriate expressions. Currently we have to change the transformed fragment manually but, in the future, we hope to be able to do that in the authoring tool itself.



```
<edge>
  <actionLabel preferPathMark="true">
    <text>[3,-2,-2,3],[Matrix0],[UpdateMatrix]</text>
    <uniqueID>3</uniqueID>

    <buggyMessage>The coefficients…</buggyMessage>
    <authorIntent>Buggy Action</authorIntent>
     …
  </actionLabel>
  <rule>
     <text>r0</text>
     <indicator>-1</indicator>
     <dimension>..</dimension>
  </rule>
  <sourceID>2</sourceID>
  <destID>3</destID>
```

```
   <rule>
     <cond buggy="3">
       <num_equal>3</num_equal>
       <num_equal>-2</num_equal>
       <num_equal>-2</num_equal>
       <num_equal>3</num_equal>
     </cond>
    <user_message
         xref="buggyMessage_2"/>
   </rule>
               . . .
<item id="buggyMessage_2">
   <material>
     <text>The coefficients…</text>
   </material>
```

Figure 4. Excerpt of the XML representation of an activity translated to the WALLIS format. The latter has to be processed manually to add missing information and to make it more general using variables.

References

1. Woolf, B., and Cunningham, P. (1987) Multiple Knowledge Sources in ITS, IEEE Expert, 2(1), 41-54.
2. Murray, T. (2003). An Overview of Intelligent Tutoring System Authoring Tools: Updated analysis of the state of the art in *Authoring Tools for Advanced Learning Environments.* Edited by Murray, T, Blessing, S. and Ainsworth S. Kluwer Academic Publishers.
3. K. Koedinger, V. Aleven, Heffernan, N.T. (2004). Opening the door to non-programmers: Authoring Intelligent Tutor Behaviour by Demonstration. 7th International Conference of ITS 2004, pp. 162-174.
4. K. Koedinger, V. Aleven, & Heffernan, N.T. (2003). Toward a Rapid Development Environment for Cognitive Tutors. *12th Annual Conference on Behaviour Representation in Modelling and Simulation.* Simulation Interoperability Standards Organization.
5. Koedinger, K. R., Suthers, D. D., & Forbus, K. D. (1999). Component-based construction of a science learning space. *International Journal of Artificial Intelligence in Education, 10*
6. M. Mavrikis (2001). Towards more intelligent and educational DGEs. Master's thesis, The University of Edinburgh, Division of Informatics; AI.
7. M. Mavrikis and A. Maciocia (2003) WaLLiS a web-based ILE for science and engineering students studying mathematics. Workshop of Advanced Technologies for Mathematics Education in 11th International Conference on AIED, Sydney.
8. C. Hunn and M. Mavrikis (2004) Improving Knowledge Representation Tutoring and Authoring in a Component Based ILE. Intelligent Tutoring Systems 2004, pp 827-829.
9. Hunn, C. (2003) Employing JESS for a web-based ITS. Master's thesis; The University of Edinburgh.
10. M.Mavrikis and A.P.Gonzalez (2003) Mathematical, interactive exercise generation from static documents in Proceedings of the MKM Symposium Edited by F. Kamarredine ENTCS 93C pp.183-201.
11. M. Kohlhase (2001) Open Mathematical Documents. Available at http://www.mathweb.org/omdoc

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

An Ontology-driven Portal for a Collaborative Learning Community

Mayorga, J.I., Barros, B., Celorrio, C., Verdejo, M.F.
Departamento de Lenguajes y Sistemas Informáticos, Universidad Nacional de Educación a Distancia, c/ Juan del Rosal, 16, E-28040
Madrid, Spain

Abstract. This paper deals with a portal for a group or community organisation and how to take advantage of having an ontology to support its operation and services. The ontology, as a representation mechanism, interprets all the available knowledge and provides adequate ways of exploiting it, which include enhanced searching capabilities to facilitate social navigation. The ontology takes into account the context and users' working style to assist the learners.

1. Introduction

Coldex Project [1] is devoted to collaborative challenge-based learning in the field of experimental sciences. A semantic portal provides support to a number of virtual learning communities, which, through this portal, do share a common body of knowledge and a variety of accessible services, which range from a Learning Object Repository (LOR, for short), storing pieces of learning material created from heterogeneous sources (knowledge providers or learning communities at work) to chat rooms. They set up a common working place where users can collaborate to carry out their assigned or chosen activities. The portal conceptual model is represented as an ontology, which includes terms and concepts such as user, group, project, activity or scenario. Users do work in a context, which is made up of a virtual workspace, the current project and activities being carried out and a social environment. For instance, the "UNED Community" could be developing the current Project, say "Organic Chemistry Analysis", being the current Activity "Identify an Element in an Infrared Spectrum". Workspaces supply an area where learners can keep their Learning Objects. They also provide links to the available services and accessible repositories, which reflect their social environment, for the sake of usability.

Learners are entitled to walk through a range of different activities, but Coldex goal, beyond mere browsing, is to foster or facilitate learning. Providing meaningful access to the available knowledge, which allows a focused information retrieval, facilitates social navigation. Ontological searching techniques allow looking for Learning Objects (LO, for short; a central element within this project) to suit users' needs. Coldex favours the concept of *thematic* objects, in the sense of LOs having been enriched via the use of ontologies [2]. Object descriptions are automatically enriched with metadata, which are generated either by the tools or taken from the context. Furthermore, these metadata are related to their ontology counterparts, so allowing the knowledge base to reflect the semantics of these annotations.

2. A case study on similarity: query patterns, context and semantics

Learners working to carry out a given task (v. gr., solving a given problem or developing a project) tend to rely on resources that have proven useful in the nearby past or that are

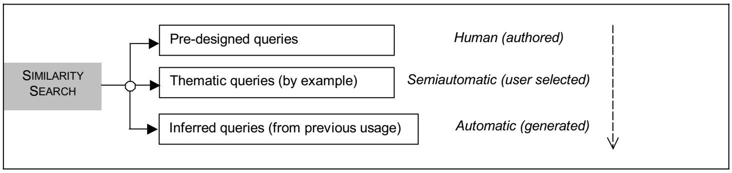

SIMILARITY SEARCH
Pre-designed queries
Thematic queries (by example)
Inferred queries (from previous usage)
Human (authored)
Semiautomatic (user selected)
Automatic (generated)

Figure 1. Different types of mechanisms to create similarity queries

somehow similar to previously useful material. Then, it is likely that users be interested in finding more or less closely related LOs, within their Learning Community, instead of wandering through the entire and potentially large knowledge base.

The rest of this section describes how we have applied Artificial Intelligence techniques for providing the user with means to fulfil this need for *similar* LOs, which are summarised in figure 1 and which range from human-authored to automatically generated:

- *Pre-designed similarity*, that is, queries written by a designer. They are likely to cover major areas, such as finding out LOs being similar because the same person has authored them. Using an ontology allows defining these queries (to compute similarity between LOs) in a declarative way, which, in turn, permits adapting them to other situations and contexts
- *Similarity by example*, i.e., looking for LOs that are like a given one [3]. The user would select a LO as the comparison pattern. This technique relates to the well-known relevance feedback concept, though the ontology extends it: the user just needs to move through the available concept network to generalise or particularise a given pattern. By deleting previously filled-in fields (or attributes, as described by the LO metadata record), the pattern would be generalised (higher abstraction). By valuing previously empty attributes, the model would be particularised (more concretion). That is, the knowledge base (with its underlying ontology) has built-in mechanisms, which permit changing the abstraction level of a query at will whilst being directed by the knowledge model
- *Inferred similarity*, which consists in exploiting information gathered from the users' work to discover new facts or regularities. This is an automatic learning mechanism. For example, users tend to visit a number of LOs while carrying out a given activity. A simple heuristic would consider these objects as being related to the mentioned activity (and user) and would assert new facts or the relationship between these entities. Further on, a query could take advantage of this knowledge, for instance, offering the user a number of LOs as linked to the original activity.

Notice that, in all the cases, the context supplies important information [6]. Besides, the queries created to exploit this kind of knowledge can be named, annotated and stored for further reference and use, either directly by the user or within the context of applicable knowledge-discovery rules.

3. Comparison with other proposals and approaches

Community portals for e-learning and Learning Object Repositories have raised a lot of interest with the result of an increasing momentum in the field and many developments, each putting the stress on their particular needs or inspirational foundation. COLDEX itself stresses the need for scenarios where learners can collaborate while carrying out experimental activities from a number of distant locations, which accounts for a distributed architecture with interconnected repositories. Furthermore, its design has favoured an ontology in order to boost knowledge-based information retrieval, on the grounds of allowing a flexible and adaptable mapping between Learning Object and the metadata annotating them as well as sharing a common understanding of the whole system.

Other efforts worth citing include POOL [4], which uses CanCore (cancore.org) as its metadata schema. POOL stress lies on the accessibility of the LOs and their accurate and flexible meta-documentation so that they can be re-used in a wide variety of educational applications. They rely on technologies such as JXTA, a peer-to-peer architecture of interconnected repositories and a network infrastructure (Ca*Net3) powering such access by means of search engines, which take profit of the metadata annotating the LOs. It is also interesting PADLR, which is related to the use of ontologies to retrieve relevant material for the learners [5].

4. Conclusions

Many efforts have been made for helping virtual learning communities to carry out learning activities. Nevertheless, many systems are more or less powerful browsers, which don't foster learning. In the field of problem-based learning, students tend to learn by doing, and to base what they do in what they or other peers have previously done. This is the reason for choosing similarity search as the case study for this paper: focused knowledge retrieval facilitates learning and even promotes it as the users find it easy to keep their own working styles, going further than a mere storing-and-browsing system. Abstraction level is an asset of this development from the representation viewpoint. The ontology allows and exploit it, as well as inference and semantic search to make possible social navigation, which we feel could play an important role in improving learning.

Acknowledgements

This work has been partially financed by the EU IST programme, COLDEX EU IST project No 2001-32327 and by Spanish Research Agency's programme TIN, project ENLACE, TIN 2004-04232

References

[1] COLDEX. URL: http://www.coldex.info

[2] Hoppe, U.; Pinkwart, N.; Oelinger, M.; Zeini, S.; Verdejo, F.; Barros, B.; Mayorga, J.I. (2005). *Building Bridges within Learning Communities through Ontologies and Thematic Objects*, to appear in CSCL 2005

[3] Pinkwart, N.; Jansen, M.; Oelinger, M.; Korchounova, L.; Hoppe, U. (2004). *Partial Generation of Contextualized Metadata in a Collaborative Modeling Environment.* In Lora Aroyo and Carlo Tasso (Eds.): Workshop proceedings of AH2004.

[4] Richards, G.; McGreal, R.; Friesen, N. (2002) *Learning Object Repository Technologies for TeleLearning: The Evolution of POOL and CanCore,* Proceedings of InSITE 2002

[5] Schmitz, C.; Staab, S.; Studer, R.; Stumme, G.; Tane, J. (2002). *Accessing Distributed Learning Repositories through a Courseware Watchdog.* Proceedings of E-Learn 2002.

[6] Stuckenschmidt, H. and van Harmelen, F. (2001). *Ontology-based metadata generation from semistructured information.* In Proceedings of K-CAP'01. Sheridan Printing.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

A Greedy Knowledge Acquisition Method for the Rapid Prototyping of Bayesian Belief Networks

Claus MÖBUS, Heiko SEEBOLD
Learning Environments and Knowledge Based Systems
Department of Computing Science, University of Oldenburg
D-26111 Oldenburg, Germany

Abstract. Bayesian belief networks (BBNs) are a standard tool for building intelligent systems in domains with uncertainty for diagnostics, therapy planning and user-modelling. Modelling their qualitative and quantitative parts requires sometimes subjective data acquired from domain experts. This can be very time consuming and stressful - causing a knowledge acquisition bottleneck.

The main goal of this paper is the presentation of a new knowledge acquisition procedure for rapid prototyping the qualitative part of BBNs. Experts have to provide only simple judgements about the causal precedence in pairs of variables. From these data a new greedy algorithm for the construction of transitive closures generates a Hasse diagram as a first approximation for the qualitative model. Then experts provide only simple judgements about the surplus informational value of variables for a target variable shielded by a Markov blanket (wall) of variables. This two-step procedure allows for very rapid prototyping. In a case-study we and two expert cardiologists developed a first 39 variables prototype BBN within two days.

Keywords. Knowledge Acquisition, Acquisition of Uncertain Causal Knowledge, Greedy Construction of DAGs in Bayesian Network Models, Greedy Construction of Hasse Diagrams and Transitive Closures, Acquisition of Causal Precedence Relations

Introduction

BBNs are relevant for the success of intelligent systems in assessing or modelling uncertain knowledge. The classical procedure for the construction of BBNs under the knowledge based approach was published by Pearl as the boundary strata method (BSM) [1]. The BSM is presented in many textbooks [2] and online tutorials. Because of its cognitive demanding aspects it is unsuitable for domain experts without modelling experience. The most problematic aspect of the procedure is the determination of a minimal set of direct influencers for a selected variable under the constraint of independence properties. Our experts had problems distinguishing between influencers and direct influencers, especially when a forgotten variable had to be included in the model again. In that case direct influencers could become indirect influencers.

This led to the development of a computerized procedure with a new greedy algorithm for the determination of transitive closures at anytime. This algorithm controls the selection of pairs, guarantees that the data comprise a partial order relation (POR) and generates the Hasse diagram of the POR (Hasse model). In the best case the monitor acquires the Hasse model of the causal precedence relation in just one pass. The savings in pair-comparisons are then (1-2/n)*100%, the judgement complexity is O(n) and the computational complexity is O(n3). If the Hasse model also passes a Markov blanket independence test, the Hasse model is without further modifications the DAG of the BBN. In the worst case the monitor needs n(n-1)/2 comparisons. The judgement complexity is O(n2) and the computational complexity stays O(n3). If the Hasse model does not pass the Markov blanket test, there is a lack of influences (or links). These must then be added back into the Hasse model. The modified DAG is then considered as the qualitative model of the BBN. Despite its flexibility, the computational complexity of the greedy algorithm is only O(n3).

The new method was successfully used to design and implement a BBN-based eLearning system for problem oriented diagnostics in aortic stenosis [3]. The knowledge acquisition for the complete model of the first prototype with 39 nodes (pair-comparisons, Markov blanket tests and estimation of conditional probability tables) could be accomplished in a two-day crash-course workshop.

A New Greedy Method for the Acquisition of DAGs in BBNs

The greediness of the new method stems from the fact that after each data input it determines which not-yet-acquired pairs are informative for the construction of Hasse diagrams. The *best case data acquisition complexity* is O(n) and the *worst case computational complexity* O(n^3).

When a pair (j, i) is presented subjects have to select a judgment from a set of alternatives {“i causes/precedes j”, “i follows j”, “i neither causes/precedes nor follows j” } internally abbreviated as {+(j,i), -(j,i), 0(j,i)}}. Though the greedy algorithm does not presuppose a special order in the data acquisition events, we selected a special order of pair comparisons along the main diagonal of the adjacency matrix. If it possible to order the variables according to some vague causal hypothesis we support the algorithm working along the main diagonal thus maximizing the number of inferences and reducing at the same time the pair comparison workload of the domain experts.

We demonstrate the algorithm with an example. First we take the DAG from Fig.1.1 as the “mental model” of the experts. Nodes are already numbered according an *ancestral ordering.*

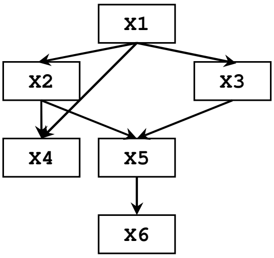


Fig. 1 - DAG of *true* model (TrM)

	1	2	3	4	5	6
1	/	$+_1$	$+_6$			
2		/	0_2	$+_7$	$+_{10}$	
3			/	0_3	$+_8$	
4				/	0_4	0_9
5					/	$+_5$
6						/

Reduction in number of pair comparisons: 33%

Tab. 1 – data acquired under *greedy* algorithm

The algorithm asks for data from the expert working above the main diagonal from top-left to bottom-right when the cell d(j,i) is empty. Diagonals move from the main diagonal in the middle of the matrix to the right upper corner. Cells are marked with “+(j,i)” (i causes/precedes j), “-(j,i)” (i follows j), “0(i,j)” (no order relation between i and j) and “/(i,j)” (transitive or reflexive cell: not necessary for Hasse diagram). Each cell entry in Tab. 1 has an index which marks the step number of the algorithm <step-nr>$_{\text{<entry>.}}$ The behaviour of the algorithm is controlled by 13 *inference rules* (Tab. 2) which are triggered after any new data entrance in cell d(i,j), and which can trigger each other by *recursive calls. The rule set is complete and can be made commutative,* if we enrich the conditions of the rules appropriately.

Nr. of rule	rule
	mirroring data and inferences
1	$+(i,j)\wedge\neg-(j,i)\Rightarrow -(j,i)$
2	$++(i,j)\wedge\neg--(j,i)\Rightarrow --(j,i)$
3	$-(i,j)\wedge\neg+(j,i)\Rightarrow +(j,i)$
4	$--(i,j)\wedge\neg++(j,i)\Rightarrow ++(j,i)$
5	$0(i,j)\wedge\neg 0(j,i)\Rightarrow 0(j,i)$
	rowwise inferences k=1,...,n
6.1	$+(i,j)\wedge+(j,k)\wedge\neg++(i,k)\Rightarrow ++(i,k)$
7.1	$+(i,j)\wedge++(j,k)\wedge\neg++(i,k)\Rightarrow ++(i,k)$
8.1	$++(i,j)\wedge+(j,k)\wedge\neg++(i,k)\Rightarrow ++(i,k)$
9.1	$++(i,j)\wedge++(j,k)\wedge\neg++(i,k)\Rightarrow ++(i,k)$
	columnwise inferences k=1,...,n
6.2	$+(k,i)\wedge+(i,j)\wedge\neg++(k,j)\Rightarrow ++(k,j)$
7.2	$+(k,i)\wedge++(i,j)\wedge\neg++(k,j)\Rightarrow ++(k,j)$
8.2	$++(k,i)\wedge+(i,j)\wedge\neg++(k,j)\Rightarrow ++(k,j)$
9.2	$++(k,i)\wedge++(i,j)\wedge\neg++(k,j)\Rightarrow ++(k,j)$

Tab. 2 - inference rules for controlling the greedy algorithm

Tab. 1 shows that we need only 10 judgements, whereas a naïve acquisition of every possible pair would take n(n-1)/2 = 15 comparisons. This 33% more efficient. Taking only the +(j,i) order information from the transitive closure, (Tab. 3) we can reconstruct the Hasse diagram (Fig. 2).

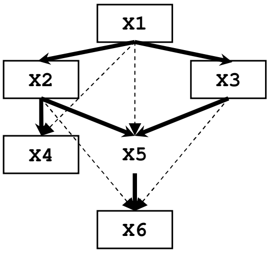


Fig. 2 - *Hasse* model reconstructed from transitive closure of input data

	1	2	3	4	5	6
1	/	+	+	++	++	++
2	–	/	0	+	+	++
3	–	0	/	0	+	++
4	––	–	0	/	0	0
5	––	–	–	0	/	+
6	––	––	––	0	–	/

Tab. 3 - transitive closure of *input* data generated by the *greedy* algorithm

References

[1] Pearl, J., 19982, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference (Revised Second Printing), Morgan Kaufman Publishers, San Mateo, CA., ISBN 0-934613-73-7

[2] Russell, St., Norvig, P., 2003, Artificial Intelligence. A Modern Approach, Upper Saddle River, N.J.: Pearson Education, Inc., ISBN 0-13-080302-2

[3] Seebold, H., Lüdtke, A., Möbus, C., 2005, Bayesian Belief Network based Diagnostics in a Problem-oriented Learning Environment for Cardiology. Proceedings of Training, Education & Simulation International 2005 (TESI 2005), Maastricht, The Netherlands, http://www.tesi2005.com/bookofproceedings/

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Automatic analysis of questions in e-learning environment

Mohamed JEMNI & Issam BEN ALI
Ecole Supérieure des Sciences et Techniques de Tunis
Research Unit of Technologies of Information and Communication (UTIC)
5, Av. Taha Hussein, B.P. 56, Bab Mnara 1008, Tunis, TUNISIA
e-mail : mohamed.jemni@fst.rnu.tn Issam.benali@laposte.net

Keywords: E-learning environment, automatic answering, latent semantic analysis, intelligent tutor.

1. Introduction

The aim of this tool is to reduce lecturers' work load and to give an immediate answer to the student (when possible) by exploring the cumulative experiences from previous students' answers for the benefit of new ones. It is well known that in every learning session of a given course, students may ask the same questions and that tutors (who may be different) may answer the same answers.

Our approach consists of storing questions/answers (with the permission of the tutor) in a data base. If any e-mail's similarity occurs regarding asked and/or answered questions, the tool tries to search for this information in the data base and answers automatically the student by giving him the stored data. Otherwise, the question will be submitted to the tutor.

2. Latent Semantic Analysis of questions

A key step of the approach consists on the semantic analysis of questions in order to compare them with others. To process this analysis, we calculate the semantic closeness between the current student's question and previous questions saved in the data base. The semantic closeness is calculated by use of Latent Semantic Analysis technique [6]. Figure 1 describes this technique.

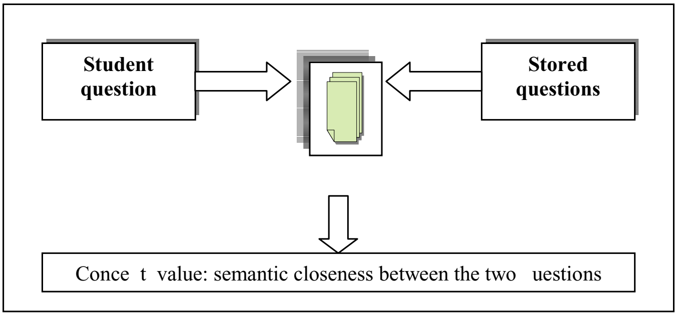


Figure 1 : Latent Semantic Analysis

The treatment of the question depends on the semantic closeness value returned by LSA.

When the analyzer does not found exactly the same question saved in the data base, the system uses other techniques. It proceeds to a refinement treatment of the collected questions when LSA could not give an accurate decision. This treatment uses the meta data introduced by the tutors who have saved the questions and their answers. Every saved question has been answered only one time when it first has been asked. In this case, the tool reacts in semi-automatic way. It may interact with the student by considering his opinion.

3. Refinement (Fuzzy treatment)

The fuzzy treatment is adopted when the system can not recognize the question with high accuracy, i.e. when LSA gives a set of candidate questions with very close semantic closeness values. In this case, the automatic answering tool proceeds in two steps:

Step 1: It proceeds to a refinement of the result by using the meta data of every question. Those meta data contain information related to the subject of the question such as the related section in the course, the cognitive level, the pre-requisite sections…
This step leads to a reduced set of candidate questions.

Step 2: The system can react with the student by proposing for him the reduced set of candidate questions and asking him to select a question from the list else to reformulate his question differently and to submit it again to the system.

Whereas if the system does not succeed to treat the student question even after the last step, the system sends the question to the tutor for answering.

4. Different phases of the treatment

To give more insight on the system's work, we can identify two different phases:

a. **Alimentation phase:** during this phase, the data base will be alimented by questions/answers under the total control of the tutors. A tutor may decide or deny saving the selected questions and their answers with an appropriate interface (figure 4).

b. **Exploiting phase:** it consists of answering automatically or semi automatically the student by using the previously stored data. Notice here, that the efficiency of the tool is ameliorated progressively as much as the data base of questions is alimented.

5. Conclusion

In this paper, we presented an automatic answering tool based on latent semantic analysis approach. The objective is to exploit the cumulative experiences from previous e-learning sessions in order to give an immediate answer to the student (when possible).
A prototype of the answering tool is currently under development. It is based on open source software i.e. MySql database, tomcat web server, JSP code and Linux operating system. We plan in the future to integrate it in the system PERSO [1] in order to experiment and evaluate it before making it available at the internet for free use.

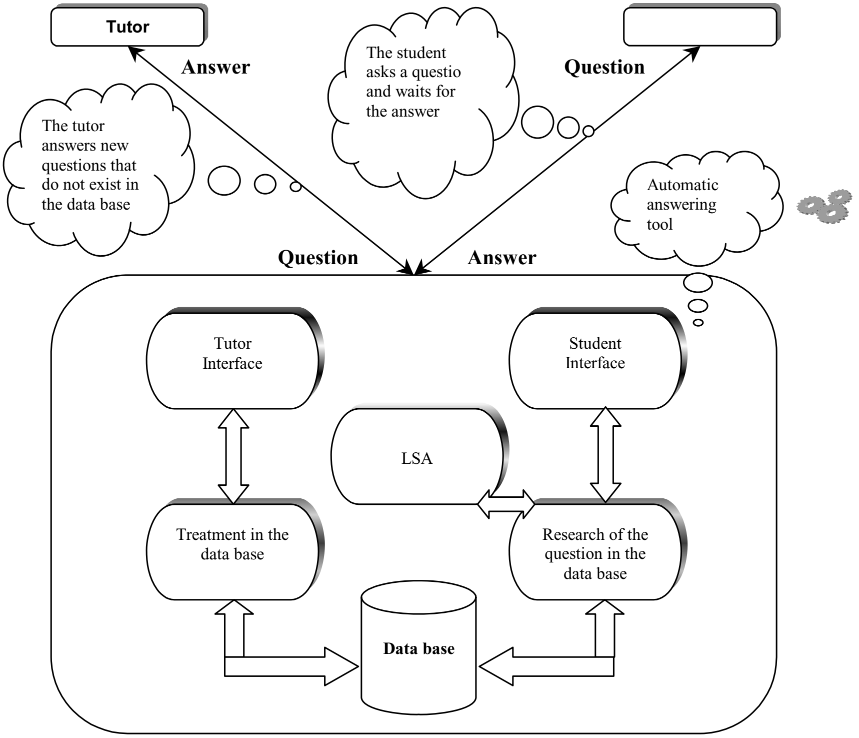

Tutor
Answer
The student asks a questio and waits for the answer
Question
The tutor answers new questions that do not exist in the data base
Automatic answering tool
Question
Answer
Tutor Interface
Student Interface
LSA
Treatment in the data base
Research of the question in the data base
Data base

Figure 2 : General architecture of the answering tool

References

[1] H. Chorfi & M. Jemni, PERSO: Towards an adaptative e-learning system, *Journal of Interactive Learning Research,* (2004), 15 (4), pp 433-447.

[2] H. Chorfi & M. Jemni, PERSO: A System to customize e-training, 5th International Conference on New Educational Environments, May 26-28 2003, Lucerne, Switzerland.

[3] P. Cotter & B. Smyth, Wapping the Web - A Case Study in Content Personalisation for WAP-Enabled Devices, In Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, Trento, Italy, August 2000.

[4] P. Gounon &B. Lemaire, Semantic Comparison of Texts for Learning Environments. In F.J. Garijo , J. C. Riquelme Santos, M. Toro (Eds): *Advances in Artificial Intelligence - IBERAMIA 2002*, Berlin: Springer Verlag LNCS 2527, 724-733.

[5] M. Jemni & I. Ben Ali, Design of an automatic answering tool for e-learning environment, poster - the International Conference on Engineering Education, October 16-21, 2004, University of Florida, Gainesville, USA.

[6] T.K. Landauer, P. Foltz, & D. Laham, Introduction to Latent Semantic Analysis, Discourse Processes, 25, pp. 259-284, 1998.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Intelligent Pedagogical Action Selection under Uncertainty

Selvarajah MOHANARAJAH, Ray KEMP, Elizabeth KEMP
Institute of Information Sciences and Technology
Massey University, New Zealand
{ S.Mohanarajah, R.Kemp, E.Kemp }@massey.ac.nz

Abstract: This paper describes a novel design for a Learner Model, which handles the effects of uncertainty formally in Pedagogical Action Selection (PAS) [1]. In our design, a mixture of Dynamic Fuzzy and Bayesian approaches are used for the PAS strategy. We treat the strength of pedagogical actions as a fuzzy variable. We also use the fuzzy logic theory effectively in dynamic prediction.

Keywords: Learner Model, Uncertainty, PAS, ITS, CBL system

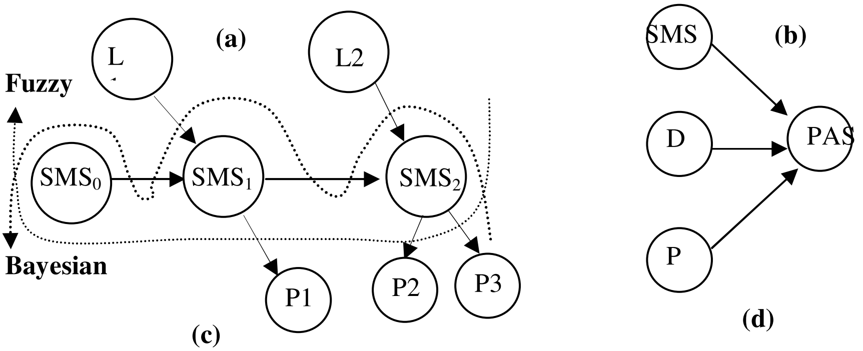
1. Introduction & Related Research

The term Pedagogical Action Selection (PAS) first coined by Mayo [1] refers to the acts of both selecting the remedy, usually feedback after a misconception is identified, and selecting the next pedagogical action. Inappropriate PAS may be generated by a CBL system due to uncertainty in its Learner Model. In this paper, we propose a unique formal approach to reduce the impact of uncertainty on pedagogical actions such as feedback and curriculum sequencing. Particularly, we use fuzzy rules to handle dynamic prediction (with a time threshold). The generic Learner Model design discussed here is domain independent and can be used for PAS in a wide range of CBL systems with minimum modification.

Recently, statistical decision theory was used for PAS in some CBL systems [1]. To the knowledge of the authors, Fuzzy Logic theory has not been used in its full strength for uncertainty handling (particularly for PAS). In this paper, we discuss a generic learner model that uses mainly fuzzy logic for PAS.

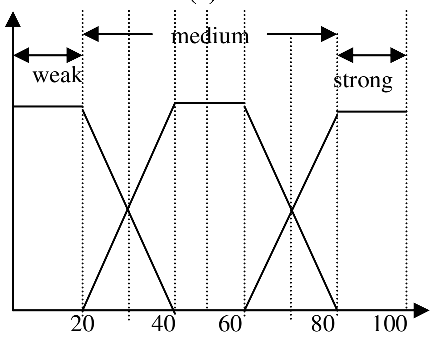
2. The Proposed Generic Learner Model

The proposed learner model is domain independent. However, the domain knowledge has to be organized in a series of main concepts. Each main concept has many sub-concepts. Each sub-concept is associated with a series of mental states (from basic to final mental state with increasing level of complexity). Each mental state is related to many questions. Naturally, the difficulty of those questions will also increase gradually from the basic to the final mental state. These questions and the relevant feedback are not assessment units, but knowledge building blocks (for e.g. scaffolding levels). After the material related to a sub concept is learned, the learner will be guided to the final mental state using relevant questions and feedback.



(d)

(c)



SMS	D	PASw	PASc
Strong	Low	PASwL1	PAScL1
Strong	Moderate	PASwL2	PAScL2
Strong	High	PASwL3	PAScL3
Medium	Low	PASwL2	PAScL2
Medium	Moderate	PASwL3	PAScL3
Medium	High	PASwL4	PAScL4
Weak	Low	PASwL3	PAScL3
Weak	Moderate	PASwL4	PAScL4
Weak	High	PASwL5	PAScL5

(e)

Level	PASw (Wrong answer): Description	PASc (Correct answer): Description
L1	Answer-Once-Again(AOA), Next-Level	Affirm, Next-Level
L2	Why-Wrong (WW), AOA, Same-Level	Why- Correct (WC), Next-Level
L3	WW& Why- Correct (WC), Same-Level	WC&Why-others-Wrong(WOW), Next-Level
L4	WW&WC, Detailed, Same-Level	WC&WOW, Detailed, Next-Level
L5	WW&WC, Detailed, Law Level	WC&WOW, Detailed, Same-Level

Figure 1: (a)&(b) Causal Relations (c)&(d) Membership& Rules (e) PAS descriptions

To incorporate dynamic adaptability, our system keeps a metric SMS, which stands for the Strength of a particular Mental State of a learner. Let SMS be 50% initially. Once material relevant for learning a concept is presented (the variable L represents the degree that a learner learned a concept in the current session), the corresponding SMS will certainly change. The variable SMS is dynamic also as its value changes over time (due to forgetting or learning from other sources, etc.) and depends on its previous values (figure 1a). In this study we will assume Markov's first order chain where the new value for SMS will depend only on its immediate preceded value. The time threshold is used to avoid updating SMS in a short duration (regardless of user interactions) [2]. Representing this dynamic causality using fuzzy model is more natural for human understanding. Besides, opening a fuzzy model is easy [3].

After learning a sub concept, the questions associated with the corresponding basic mental state will be presented. We let the variable P stand for a learner's performance level in a question. For simplicity, we will assume that the questions are traditional multiple-choice questions (MCQ), and therefore, P will not be fuzzy and will take two values; true or false. The learner's performance depends on their mental states (however, lucky guesses or careless mistakes can also play a role). We let a fuzzy variable D represent the difficulty level of a question, which takes three values: Difficult, Moderate and Easy. We assume the fuzzy membership model in figure 1(c) for the variable SMS (and also for L and D).

3. Intelligent Pedagogical Action Selection

We keep two disjoint sets of fuzzy variables for the strength of pedagogical actions, one for correct answers (PASc) and the other for wrong answers (PASw) (figure 1(e) - we hope the descriptions given are sufficient to grasp their meaning). Both variables vary from one to five levels with increasing pedagogical strength. The appropriate pedagogical action depends on three variables: the strength of the mental state (SMS), difficulty level of the question (D), and performance (P) (figure 1(b)). The table in figure 1(c) gives the relevant fuzzy rules. For PAS process, assume SMS and P are independent.

Firstly, in the fuzzification process, the corresponding fuzzy values are obtained for the crisp input for difficulty level of a question (D) using fuzzy membership function in figure 1(c) (the SMS will already have fuzzy values usually). Thereafter, appropriate fuzzy rules (figure 1(d)) will be applied to obtain the fuzzy output values for PAS. Finally, the defuzzification process determines the crisp value of PAS. Simply the PAS variable with highest probability can be selected (If two PAS levels have equal beliefs the PAS with highest strength will be selected). Otherwise, to get a unique numeric representation for PAS, we could use Larsen's Product Rule combined with mirror rule at extremes for defuzzification process [4]. Finally, the SMS will be updated using Bayesian rules using performance (P) as evidence (figure 1(a)).

4. Limitations & Future work

The success of this learner model heavily depends on the level of detail in the domain model. The potential dependencies between various mental states need to be identified. The heuristic estimates and some weak assumptions (e.g. time length does not affect learning) need to be revised. Finally, we have just completed a functional prototype of this learner model for our CBL system LOZ. We plan to evaluate this prototype in the forthcoming semester.

References

[1] M. Mayo and A. Mitrovic, "Optimising ITS behaviour with Bayesian networks and decision theory," *International Journal of AI in Education (IJAIED)*, vol. 12, pp. 124-153, 2001.

[2] J. Reye, "Student Modeling based on Belief Networks," *International Journal of Artificial Intelligence in Education (IJAIED)*, vol. 14, 2004.

[3] S. Mohanarajah, R. Kemp, H., and E. Kemp, H., "Unfold the scaffold & Externalizing a Fuzzy Learner Model," presented at ED-MEDIA05 (To be appeared), Montreal, Canada, 2005.

[4] A. A. Hopgood, *Intelligent systems for engineers and scientists*, 2nd ed. Boca Raton, FL: CRC Press, 2000.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

A Generic Tool to Browse Tutor-Student Interactions: Time Will Tell!

Jack Mostow, Joseph Beck, Andrew Cuneo, Evandro Gouvea, and Cecily Heiner
Project LISTEN (www.cs.cmu.edu/~listen), Carnegie Mellon University
RI-NSH 4213, 5000 Forbes Avenue, Pittsburgh, PA. USA 15213-3890

Abstract. A basic question in mining data from an intelligent tutoring system is, "What happened when…?" A generic tool to answer such questions should let the user specify which phenomenon to explore; explore selected events and the context in which they occurred; and require minimal effort to adapt the tool to new versions, to new users, or to other tutors. We describe an implemented tool and how it meets these requirements. The tool applies to MySQL databases whose representation of tutorial events includes student, computer, start time, and end time. It infers the implicit hierarchical structure of tutorial interaction so humans can browse it. A companion paper [1] illustrates the use of this tool to explore data from Project LISTEN's automated Reading Tutor.

1. Introduction

Intelligent tutoring systems' ability to log their interactions with students poses both an opportunity and a challenge. Compared to human observation of live or videotaped tutoring, such logs can be more extensive in the number of students, more comprehensive in the number of sessions, and more exquisite in the level of detail. They avoid observer effects, cost less to obtain, and are easier to analyze. For example, Project LISTEN's Reading Tutor listens to children read aloud, and helps them learn to read [2]. A Reading Tutor session consists of reading a number of stories. The Reading Tutor displays a story one sentence at a time, and records the child's utterances for each sentence. The Reading Tutor logs each event (session, story, sentence, utterance, …) into a database table for that event type. Data from tutors at different schools flows into an aggregated MySQL database server [3]. Our 2003-2004 database includes 54,138 sessions, 162,031 story readings, 1,634,660 sentences, 3,555,487 utterances, and 10,575,571 words. Such data is a potential gold mine [4].

Mining such data requires the right tools to locate promising areas, obtain samples, and analyze them. One part of this process is in-depth qualitative analysis of individual tutorial events. Such case analyses serve various purposes, for example:

- Spot-check tutoring sessions to discover undesirable tutor-student interactions.
- Identify the most common types of cases in which a specified phenomenon occurs.
- Formulate hypotheses by identifying features that examples suggest are relevant.
- Sanity-check a hypothesis by checking that it covers the intended sorts of examples.

This paper describes a tool, implemented as a Java™ program that queries MySQL databases, that supports such case analysis by exploiting three simple but powerful ideas. First, a student, computer, and time interval suffice to specify an event. Second, a containment relation between time intervals defines a hierarchical structure of tutorial interactions. Third, the first two ideas make it possible to implement a generic but flexible tool for mining tutor data with minimal dependency on tutor-specific details.

2. Specify which phenomenon to explore.

First, how can we specify events to explore? A deployed tutor collects too much data to look at, so the first step in mining it is to select a sample. A database query language provides the power and flexibility to describe and efficiently locate phenomena of interest.

For example, the query "`select * from utterance order by rand() limit 10`" selects a random sample of 10 from the table of student utterances. Whether the task is to spot-check for bugs, identify common cases, formulate hypotheses, or check their sanity, our mantra is "check (at least) ten random examples." Random selection assures variety and avoids the sample bias of, for example, picking the first ten examples in the database.

Although an arbitrary sample like this one is often informative, a query can focus on a particular phenomenon of interest, such as: Which questions did students take longest to answer? Or: When did students get stuck long enough for the Reading Tutor to prompt them? Exploring examples of such phenomena can help the researcher spot common features and formulate causal hypotheses to test with statistical methods on aggregated data.

Second, what information suffices to identify a tutorial interaction? A key insight here is that student, computer, and time interval are sufficient, because together they uniquely specify the student's interaction with the tutor during that time interval. (We include computer ID in case the student ID is not unique.) This "lowest common denominator" should apply universally to virtually any tutor.

Third, how can we translate the result of a query into a set of tutorial events? The tool scans the labels returned as part of the query, and finds the columns for student, computer, start time, and end time. The code assumes particular names for these columns, e.g. "`user_id`" for student, "`machine_name`" for computer, and "`start_time`" for start time. If necessary the user can enforce this naming convention, e.g., by inserting "`as start_time`" in the query to relabel the column. We require that the fields for student, computer, start time, and end time be keys in the database tables. Indexing tables on these fields enables fast response by the tool even for tables with millions of records.

3. Explore selected events and the context in which they occurred.

What context frames an event? Our answer is: "its chain of ancestor events." *E.g.*, the context of a word includes the utterance, sentence, story, and session in which it was read.

How can we discern the hierarchical structure of student-tutor interaction? At first we computed this hierarchy using its hardwired schema for the Reading Tutor database to determine which events are part of which others. But then we had a key insight: exploit the nested time intervals in the data logged by our tutor – and probably by many others too.

If events A and B have the same student and computer, when is A an ancestor of B? We initially required that A contain all of B. But we relaxed the criterion to better handle occasional overlapping intervals in our data. We therefore define A as an ancestor of B if B starts during A. Thus a word's ancestors include an utterance, sentence, story, and session.

The tool computes the event tree by partial-ordering the events according to the ancestor relation. The parent of an event is defined as its minimal ancestor. Siblings are defined as sharing the same parent; they are ordered by their start times.

The companion paper [1] shows how the tool displays such trees, summarizes events in readable form, and lets users dynamically drill down and adjust which details to display.

4. Require minimal effort to adapt the tool to new versions, new users, or other tutors.

How can the tool obtain the information it needs about a database of tutor interactions? Its generic architecture enables it to make do with readily available meta-data, a few assumed conventions, and a little code. MySQL provides the required meta-data, namely the list of tables in the database, the fields in each table and event list, and their names and data types. We exploit the observation (or assumption) that the meaning of field names is consistent across database tables and over time. The code assumes particular field names for student, machine, and start and end times, but overrides this convention when necessary, as in the case of a particular table with a "`Time`" field instead of a "`Start_time`" field.

The method to compute the context of a selected target event is: First, extract its student, computer, and start time. Then query every table of the database for records for the same student and computer whose time interval contains the start of the target event. Finally, sort the retrieved records according to the ancestor relation, and display them accordingly by inserting them at appropriate positions in a Java™ expandable tree widget.

The method to find the children of a given event fires only when needed to expand the event node. It finds descendants in much the same way as the method to find ancestors, but then winnows them down to the children (those that are not descendants of others). Both methods work whether the events are in the same table or in different tables.

A more knowledge-based method would know which types of Reading Tutor events can be parents of which others. However, this knowledge would be tutor- and possibly version-specific. In contrast, our brute force solution of querying all tables requires no such knowledge. Moreover, its extra computation is not a problem in practice. Our databases consist of a few dozen tables, the largest of which have tens of millions of records. Despite this table size, the tool typically computes the context of an event with little or no delay.

5. Conclusion

This paper reports an implemented, efficient, generic solution to a major emerging problem in educational data mining: efficient exploration of vast student-tutor interaction logs. We describe three useful requirements for such exploration that an earlier tool [5] failed to meet, and how the new tool meets them: let the user specify which phenomenon to explore; explore selected events and the context in which they occurred; and require minimal effort to adapt the tool to new versions, to new users, or to other tutors. Our key conceptual contribution uses temporal relations to expose natural hierarchical structure. This is the sense in which "time will tell" many basic relationships among tutorial events.

The success of this approach suggests specific recommendations in designing databases of tutorial interactions: Log each distinct type of tutorial event in its own table. Include student ID, computer, start time, and end time as fields of each such table so as to identify its records as events. Name these fields consistently within and across databases created by successive versions of the tutor so as to make them easier to extract.

The ultimate test of this tool is whether it leads to useful discoveries, or at least sufficiently facilitates the process of educational data mining that the miners find it helpful and keep using it. To repeat our subtitle in its more usual sense, "time will tell!"

Acknowledgements: This work was supported in part by the National Science Foundation under ITR/IERI Grant No. REC-0326153. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation or the official policies, either expressed or implied, of the sponsors or of the United States Government. We thank the educators and students who generated our data.

References (see www.cs.cmu.edu/~listen)

1. Mostow, J., J. Beck, H. Cen, E. Gouvea, and C. Heiner. Interactive Demonstration of a Generic Tool to Browse Tutor-Student Interactions. *Supplemental Proceedings of the 12th International Conference on Artificial Intelligence in Education (AIED 2005)* 2005. Amsterdam.
2. Mostow, J., G. Aist, P. Burkhead, A. Corbett, A. Cuneo, S. Eitelman, C. Huang, B. Junker, M.B. Sklar, and B. Tobin. Evaluation of an automated Reading Tutor that listens: Comparison to human tutoring and classroom instruction. *Journal of Educational Computing Research*, 2003. *29*(1): p. 61-117.
3. MySQL. Online MySQL Documentation. 2004.
4. Beck, J., ed. *Proceedings of the ITS2004 Workshop on Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes*. 2004: Maceio, Brazil.
5. Mostow, J., J. Beck, R. Chalasani, A. Cuneo, and P. Jia. Viewing and Analyzing Multimodal Human-computer Tutorial Dialogue: A Database Approach. *Proceedings of the ITS 2002 Workshop on Empirical Methods for Tutorial Dialogue Systems*, 75-84. 2002. San Sebastian, Spain.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Effects of Dissuading Unnecessary Help Requests While Providing Proactive Help

R. Charles MURRAY and Kurt VANLEHN
Learning Research and Development Center, University of Pittsburgh
{rmurray,vanlehn}@pitt.edu

Abstract. We tested effects of dissuading students from requesting help unless they really needed it. The manipulation occurred while the students solved problems on an ITS that provided proactive help. Compared to their counterparts, *dissuaded students* requested help much less often. Moreover, the less help students requested, the higher their posttest score. Among students with lower pretest scores, dissuaded students marginally gained more than their non-dissuaded counterparts. We discuss our results, a new type of help abuse, some ramifications of proactive help, and the generalizability of our results.

Introduction

A scourge of the ITS field is that students often misuse help – for instance, using help excessively to avoid thinking or learning, or under-using help when they need it [1, 2]. The reasons for help-seeking behaviors can be complex [3] and so influencing them is likely to be complex as well. Empirical information about the effects of interventions can be used to begin to unravel the complex relationships among the factors that influence help-seeking behavior.

We tried a small set of easy-to-implement interventions intended to dissuade students from asking for help when they did not need it and thereby to improve their learning. Featured among these was a delay before students received requested help. It is widely thought that help delay might reduce help requests and improve learning, which is why Cognitive Tutors, used in hundreds of schools, now delay before they provide bottom-out help (K.R. Koedinger, personal communication, 2005). However, we know of no research that tests such hypotheses.

We implemented these interventions in an ITS that expands upon the traditional reactive role of computer tutors by providing unsolicited, *proactive* help as well as *reactive* responses to help requests. Our Calculus Tutor [4] attempts to look ahead to anticipate a student's need for help, just as human tutors often do [5], rather than always waiting for the student to request help or make an error. This experiment was conducted during a data collection phase when the Calculus Tutor did not use its decision-theoretic capabilities but instead selected randomly from relevant help messages and randomly chose to provide proactive help on about 50% of its opportunities. The Calculus Tutor considers providing proactive help (1) when the student selects an uncompleted step to work on, and (2) when the student makes an error.

1. Experimental Manipulation

Only students who had not yet encountered the domain material covered by the Calculus Tutor participated in the study. These students were randomly assigned to one of two conditions with 27 students in the experimental condition and 33 students in the control condition. The only difference between the two conditions was a set of three interventions in the experimental condition: (1) students were asked not to request help unless they really needed it; (2) students were told that there would be a "substantial delay" when they asked for help; and (3) the Calculus Tutor delayed 10 seconds before it provided reactive help. For the control condition, students were not told anything about requesting help or a help delay and the Calculus Tutor did not delay before providing reactive help. Before using the tutor, all students studied a printed tutorial for about 45 minutes and then took a 28-item pretest. The tutorial and pretest covered all of the main concepts required to solve the Calculus Tutor's problems. Students then used the Calculus Tutor to solve the same 5 multi-step problems. Afterwards, all students took a posttest which was isomorphic to the pretest. There were no significant differences in the groups' pretest scores or time on task.

2. Effects

2.1 Dissuaded students requested help less often

The mean number of help requests for students who were dissuaded from requesting help (*dissuaded students*) was 7.4, while the mean number of help requests for *non-dissuaded students* was 22.8. This difference was significant, $t(58)=2.43$, $p<.01$ (1-tailed).

Therefore, dissuaded students requested help significantly less often.

2.2 For all students, number of help requests was negatively correlated with posttest scores

For students in both conditions, pretest and number of help requests accounted for 61% of the variance in predicting posttest scores, adjusted $R^2=.611$, while pretest alone accounted for 41% of the variance, adjusted $R^2=.407$, a difference of 20%. ANOVAs for the ratios of variances explained by both models were significant, $p<.001$. The standardized Beta coefficient for number of help requests was -.472 and this was significant, $t = -5.163$, $p<.001$.

Thus, students who requested help less often scored higher on the posttest, whether or not they were dissuaded.

2.3 Among students with lower pretest scores, dissuaded students marginally gained more

Overall, there was no significant difference between dissuaded and non-dissuaded students in learning gain. To test for an aptitude-treatment interaction, we divided the students using a median split based on pretest scores. The median pretest score was 19 and the *low pretest* group consisted of 28 students with scores ranging from 8 to 18, of which half were dissuaded. Among low pretest students, the mean net gain for dissuaded students was 4.9 out of 28, while the mean net gain for non-dissuaded students was 1.8. Measuring learning gain (LG) using the formula LG = (posttest % – pretest %) / (100% – pretest %), this difference was marginally significant, $t(26)=1.713$, $p=.099$ (2-tailed).

While our intervention didn't help all students, it may have helped those with the most to gain, the students with lower scores on the pretest.

3. Discussion and Conclusions

The intervention was successful at dissuading students from requesting help. Our intention was to motivate students to strive to make connections between their existing knowledge and the task at hand before requesting help, and then if they really needed to request help, to try their best to learn from the help that they received before requesting help again. We hypothesized that such behavior would lead to more learning and thus to higher gains and posttest scores. Supporting our hypothesis, we found a strong negative correlation between number of help requests and posttest scores. However, we found that only dissuaded students with lower pretest scores marginally gained more than their non-dissuaded counterparts. We suspect that this is the result of a classic aptitude-treatment interaction. Higher pretest students likely had better learning skills. Students with better learning skills are more likely to use help appropriately – e.g., when they really need it. Thus, dissuading higher pretest students from using help except when they really needed it had little effect on their behavior – they didn't need to be dissuaded – or their posttest scores. In contrast, non-dissuaded students with poorer learning skills or less motivation to learn were more likely to avoid engaging with the material by requesting help until the tutor gave the answer away.

We observed a few instances of a new type of help abuse that may have been elicited by dissuading help requests while providing proactive help when the student selects a step: repeatedly selecting a step and then canceling it until proactive help is received. This is similar to making frivolous errors when proactive help for errors is available. The decision-theoretic version of the Calculus Tutor incorporates a model of the student's manner of help usage partly to recognize and counteract such behaviors.

Our results are linked to proactive help because students were able to receive help without requesting it. We hope that with more carefully considered proactive help (e.g., decision-theoretic), dissuading unnecessary help requests while providing proactive help will be a step towards changing students' use of ITS help. Currently, help requests are usually the main way for students to get an ITS to respond, but these help requests are often either over- or under-utilized, resulting in sub-optimal learning. If students are dissuaded from requesting help when they don't need it, and provided with proactive help when they do need it, students may begin to request help from ITSs more like they request help from human tutors: when they really need it and the tutor has failed to anticipate their need.

Since our intervention involved mainly a simple help delay, it can be used by tutors in diverse domains with a variety of help content. The delay was only for reactive help and so it may prove useful even for tutors that do not provide proactive help.

References

[1] Aleven, V., B.M. McLaren, & K.R. Koedinger (2004). Toward Tutoring Help Seeking: Applying Cognitive Modeling to Meta-Cognitive Skills. *Seventh International Conference on Intelligent Tutoring Systems, ITS 2004.*

[2] Baker, R.S., A.T. Corbett, & K.R. Koedinger (2004). Detecting Student Misuse of Intelligent Tutoring Systems. *Seventh International Conference on Intelligent Tutoring Systems, ITS 2004.*

[3] Aleven, V., E. Stahl, S. Schworm, F. Fischer, & R.M. Wallace (2003). Help Seeking in Interactive Learning Environments. *Review of Educational Research*, 73(2), 277-320.

[4] Murray, R.C., K. VanLehn, & J. Mostow (2004). Looking ahead to select tutorial actions: A decision-theoretic approach. *International Journal of Artificial Intelligence in Education*, 14(3-4), 235-278.

[5] Merrill, D.C., B.J. Reiser, S.K. Merrill, & S. Landes (1995). Tutoring: Guided learning by doing. *Cognition and Instruction*, 13(3), 315-372.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Breaking the ITS Monolith: a Hybrid Simulation and Tutoring Architecture for ITS

William R. MURRAY
Teknowledge Corporation,
1800 Embarcadero Road, Palo Alto 94303, USA

Abstract. The classic ITS conceptual diagram with boxes for domain, user, and pedagogical modeling encourages a monolithic architecture. Furthermore, it focuses on knowledge and simulation of tutoring capability while deemphasizing knowledge and simulation of the task environment. However, the actual task environment may require complex domain and graphical simulations whose software investment vastly exceeds the ITS, even though these simulations are commingled with other aspects of the ITS in the classic box-diagram view.

A more useful architecture for combining intelligent tutoring capabilities with complex pre-existing simulations (e.g., tactical simulations) views the ITS system as a tutor-simulation hybrid. The ITS consists of two communicating components, a simulation agent and a tutor agent. The tutor agent models the pedagogical capabilities of an expert instructor. The simulation agent is the simulation wrapped to send an event stream to the tutor, and to send and receive requests.

This hybrid architecture breaks up an ITS monolith into two reusable components. It allows the tutor agent to be reused in multiple teaching applications with different simulations and vice versa. The V-CTC system, which provides a reusable tutoring component to provide ITS capabilities for tactical simulations, illustrates this architecture.

Introduction

The V-CTC system [1] is an example of the hybrid simulation and intelligent tutoring system (ITS) architecture shown in Figure 1. The ITS system ties together a simulation with a tutor component via an event stream and software connectors [2]. The simulation, formerly standalone, is modified to provide an event stream to detail events internal to the simulation model. Software connectors, which trap standard DLL calls (e.g., to GUI toolbox elements), detail events that are graphically visible and do not require developer modifications. The two streams are collated within the tutor component. The modified simulation is called the simulation component or just “the sim” in the hybrid architecture.

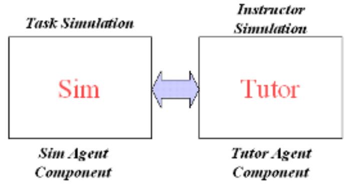


Figure 1. The Hybrid Sim-Tutor Intelligent Tutoring System Architecture used in V-CTC

The tutor component is a reusable software component built over an ontology for combined-arms warfare. It has domain-general tutorial strategies culled from a subject matter expert (SME) at the National Training Center at Fort Irwin, California. This component provides ITS capabilities when coupled to a tactical simulation, but is not intended to run standalone. The tutor component uses a blackboard architecture to mediate events from the simulation and requests for information from the tutor component. Knowledge sources (KSes) track user progress in the simulation and trigger user model updates, tutor interventions, and after-action reviews (AARs).

The approach in the V-CTC project is to combine previously standalone PC-based high-fidelity tactical simulations, such as Arm
intelligent tutoring component. A SME developing their own simulation will typically not provide an API. Instead, their effort is spent on improving tactical AI, adding new weapons, scenarios, and maps, etc. The V-CTC tutor component leverages its mathematical and graphical simulations; its set of scenarios; its scenario editor; and its tactical AI (friendly and enemy).

The ITS component provides the sim with an instructor capability that can intervene during the simulation and provide an AAR. The Army refers to the instructor personnel at its combat training centers (CTCs) as observer / controllers (O/Cs). V-CTC plays the role of a virtual O/C operating in a simulated CTC provided by the tactical sim. The sim provides quantitative feedback, while the tutor provides a complementary role: it provides qualitative feedback. [1].

Figure 1 shows the abstract view of V-CTC‘s hybrid ITS-sim architecture. The ITS and sim are tied together through communication channels. Events flow primarily from the sim to the ITS so the ITS can track the trainee's progress in problem-solving. Messages back from the ITS to the sim tell it to pause or take actions such as highlighting units or loading scenarios.

In its first application as a prototype tutor for battalion fire support officers (BN FSOs), V-CTC uses a developer-adapted version of the simulation Armored Task Force (ATF) as the sim part of the tutor-sim hybrid. 6 missions are currently implemented. An example of a tutorial intervention (guidance), shown in Figure 2, recommends firing at higher-priority targets.

2. A Hybrid Tutor-Simulation Architecture: The Virtual Combat Training Center

V-CTC's hybrid architecture views an ITS system as a coupling of two separate simulations: one simulation for the problem-solving environment (the sim) and one simulation for the instructor who operates in that environment (the tutor). Each simulation is a separate component that is also an agent as it sends and receives messages from the other and performs actions both autonomously and in response to requests from the other.

Clancy's GUIDON [6] research shows that additional knowledge is needed for instruction beyond expert domain knowledge, and this is true in V-CTC, too. For example, to track the trainee's progress it is important to know what maneuver phase the friendly forces are in, and this, in turn, depends on where the lead maneuver unit is, but the sim does not track this normally.

The V-CTC tutor component must be able to handle a real-time stream of messages arriving from the sim and respond with real-time requests of its own. Time latency can be critical. For example, suppose a fire mission occurs. After receiving this event, the tutor component immediately sends a request for the set of all visible targets to score the fire mission. If it waits too long the targets change: They may be destroyed, or move out of visible range.

A blackboard architecture was a good fit for the tutor component as it handles real-time events, both internal and external. External real-time events come from the sim. Internal real-time events are timer events that trigger query messages sent to the sim to monitor user progress, such as the location of the lead maneuver unit. Events from either trigger KSes. The blackboard architecture is based on the blackboard system BB1 Version 2.1 [7].

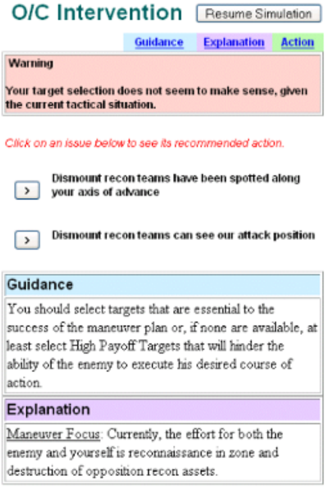
3. Related Work

Steve Ritter's Plug-In ITS [8,9] proposes an ITS component that plugs into applications primarily to use them as interfaces or displays, or to add ITS functionality. In contrast, V-CTC focuses on simulations and providing a reusable ITS component with an ontology, and tutoring strategies, all targeted to a specific domain class (tactics). GUIDON allows a rule-based expert system to become an ITS with the addition of tutorial rules to simulate an instructor. Non-rule-based tactical simulations such as ATF or the simulations of physical phenoma used in RBT [10] and SOPHIE [11] are not handled. RBT and SOPHIE, in turn, are ITSs incorporating simulations, but each is not intended to be generic: each does not have separable, reusable, domain-general ITS-components. Finally, SHAI's BC2010 ITS [13] couples a pre-existing standalone ITS to the BC2010 simulation. It critiques a completed plan or execution run when requested rather than allowing tutorial pop-ups at the tutor's judgement; thus it is less tightly coupled and less interactive than V-CTC. Also, there is no ontology, and the tutorial strategies are generic, not specific to tactical simulations.

4. Conclusions

The hybrid architecture provides significant cost and labor savings by leveraging a high-fidelity simulation developed by a SME over a period of years. The tutor developers can not only use the underlying tactical simulation, but also the simulation's GUI, the existing set of scenarios, and the scenario editor.

The hybrid architecture provides a different perspective in ITS architecture: one of two coupled components, agents, or simulations (task + tutor); compared to a single monolithic ITS of the classical ITS architecture. In the latter view an ITS should prevent all false alarms leading to bad advice, or it loses credibility. In contrast, the hybrid architecture's components need not be perfect if users understand their roles as task + tutor simulations.



Warning

Your target selection does not seem to make sense, given the current tactical situation.

Click on an issue below to see its recommended action.

Dismount recon teams have been spotted along your axis of advance

Dismount recon teams can see our attack position

Guidance

You should select targets that are essential to the success of the maneuver plan or, if none are available, at least select High Payoff Targets that will hinder the ability of the enemy to execute his desired course of action.

Explanation

Maneuver Focus: Currently, the effort for both the enemy and yourself is reconnaissance in zone and destruction of opposition recon assets.

Fig 2. Example of V-CTC guidance

The misleading aspect of the classical view is that these knowledge-based capabilities (domain, user, tutoring) appear wrapped in one container. It suggests that the system should be a single agent architecture loading multiple knowledge bases. We propose instead that the architecture may be usefully broken down into two primary components (coupled agents / simulations) when we are adding intelligent tutoring capabilities to pre-existing simulations: a knowledge-based tutoring component, with many of the capabilities of an ITS, but not necessarily operational standalone; and a wrapped or modified version of the simulation that can interact with the ITS component, and which may have unique AI capabilities of its own.

References

[1] William R. Murray, and Michelle Sams: Virtual Combat Training Center (V-CTC): An Intelligent Tutoring System + Tactical Simulation**.** In *I / ITSEC.* 2004.

[2] Robert M. Balzer and Neil M. Goldman. Mediating Connectors: A Non-ByPassable Process Wrapping Technology. In Proceeding of the 19th IEEE International Conference on Distributed Computing Systems.

[3] ProSimCo. www.prosimco.com Description of Armored Task Force.

[5] Battlefront.com. www.battlefront.com Description of TacOps.

[6] W. Clancey. Knowledge-Based Tutoring. The GUIDON Program. MIT Press. Cambridge, MA. 1987.

[7] Barbara Hayes-Roth. An Architecture for Control. Artificial Intelligence. Vol 26,3 (1985). Pp:251 – 321.

[8] Steven Ritter, Peter Brusilovsky, Olga Medvedeva: Creating More Versatile Intelligent Learning Environments with a Component-Based Architecture. *Intelligent Tutoring Systems 1998*: 554-563

[9] S. Ritter, K Koedinger: An Architecture For Plug-In Tutor Agents. In *AI and ED* 7(3/4): 315-347 (1996).

[10] B.Woolf,D.Blegen, J.Jansen, A.Verloop:Teaching a Complex Industrial Process. *AAAI 1986*:722-729

[11] John Seely Brown, Richard R. Burton, Alan G. Bell: SOPHIE: A Step Toward Creating a Reactive Learning Environment. *International Journal of Man-Machine Studies 7*(5): 675-696 (1975)

[12] Richard H. Stottler, Randy Jensen, Bill Pike, Rick Bingham : Adding An Intelligent Tutoring System To An Existing Training Simulation. In *I/ITSEC.* 2002.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

A Study on Effective Comprehension Support by Assortment of Multiple Comprehension Support Methods

Manabu NAKAMURA, Yoshiki KAWAGUCHI, Noriyuki IWANE,
Setsuko OTSUKI, and Yukihiro MATSUBARA
Faculty of Information Sciences, Hiroshima City University,
3-4-1, Ozuka-higashi, Asaminami-ku, Hiroshima-shi, 731-3194 Japan

Abstract. This paper proposes an effective comprehension support by an assortment of multiple comprehension support methods. Each comprehension support method has strong and weak points. However, comprehension support methods can play complementary roles. These methods can be combined so that these methods may play complementary roles and more effective comprehension support can be achieved by this combination. We constructed a compound comprehension support system by assorting multiple comprehension support methods. The evaluation of this system is given in this paper.

1. Introduction

So far, various comprehension support methods have been studied [1, 2]: CAI (Computer Assisted Instruction), ITS (Intelligent Tutoring System), ILE (Interactive Learning Environment), CSCL (Computer Supported Collaborative Learning), Simulation, Animation, Movie, Figure, Text, and so on. Each of them has its own strong and weak points. However, comprehension support methods can play complementary roles. An assortment of multiple comprehension support methods is needed to supplement weak points of a comprehension support method. Learners can use multiple comprehension support methods if need.

This paper proposes an effective comprehension support by an assortment of multiple comprehension support methods. In this study, we constructed and evaluated a compound comprehension support system about pendulums by assorting multiple comprehension support methods. We verified strong and weak points of each comprehension support method and effectiveness of an assortment of multiple comprehension support methods. An assortment of multiple comprehension support methods is effective for many-sided and overall comprehension. The features of comprehension support methods are described in section 2. The compound comprehension support system is described in section 3. The evaluation of the compound comprehension support system is described in section 4. Finally, conclusions are given in section 5.

2. Features of Comprehension Support Methods

Each of comprehension support methods has its own strong and weak points. We describe own strong and weak points of (1) Text, (2) ILE, and (3) ITS in this section. A learner can learn systematically by using (1) Text. He/she can acquire unknown knowledge with ease. He/she can use it like a dictionary by using hyperlinks and its index. On the other hand, he/she may not be interested in passive learning only by (1) Text. It is difficult for him/her to image the target if the target is dynamic. (2) ILE has simulators, which use multimedia technology (e.g. graphics, animation, and so on). It is intuitive and easy to understand. It supports discovery and creative learning. He/she learns by operating the target directly and observing the change of state. It can visualize invisible natural phenomena (e.g. a force, a sound, and a velocity) and mathematical concepts. Furthermore, he/she can reconstruct his/her own mental model independently by repetition of hypotheses constructs, experiments, and verifications. On the other hand, it can't give him/her advices and answer his/her questions when he/she reaches an impasse. (3) ITS guesses his/her problem solving process, constructs his/her comprehension state model, and supports his/her comprehension. It can infer correct answer and adopt explanations to his/her answers and questions. On the other hand, it is difficult to support his/her comprehension always suitably because his/her questions become over the prepared knowledge.

3. Compound Comprehension Support System

There are fields that each of comprehension support methods can support effectively and fields that each of comprehension support methods can't support effectively. Thus, it is difficult for each of comprehension support methods to support the field that it can't support effectively. A learner can reach an impasse. It is desirable that he/she can use multiple comprehension support methods to avoid this impasse. For example, a learner may not be interested in passive learning only by (1) Text. (2) ILE can play the complementary role because it is a learner-initiative comprehension support method. However, it can't give him/her advices and answer his/her questions when he/she reaches an impasse. (3) ITS can play the complementary role because it can answer his/her questions. However, it can answer his/her questions only in the prepared knowledge. (2) ILE can play the complementary role because he/she can reconstruct his/her own mental model independently by repetition of hypotheses constructs, experiments, and verifications.

This paper proposes an effective comprehension support by an assortment of multiple comprehension support methods. We constructed a compound comprehension support system about pendulums [3], that has multiple comprehension support methods in order to verify strong and weak points of each comprehension support method and effectiveness of an assortment of multiple comprehension support methods. The compound comprehension support system has a text about a pendulum, a microworld with simulators of pendulum, and a question and answer system by QSIM (Qualitative SIMulation) and DQ analysis (Differential Qualitative analysis).

(1) Text about a pendulum shows the definition of x (the position of weight), v (the velocity of weight), a (the acceleration of weight), and T (the period of motion of weight), formulas of them, and a graph of v (the velocity of weight) and a (the acceleration of weight). (2) Microworld is an implementation of (2) ILE's features and has multiple simulators of pendulum. A learner can change θ (the initial position (angle) of weight), l (the length of string), m (the weight of weight), and/or g (the gravity acceleration) and observe the motion of pendulum. a (the acceleration of weight), v (the velocity of weight), mg (the gravity force of weight), mgcosθ (the tension of weight) mgsinθ (the driving force of weight) aren't easy to observe in real but can be visualized in this system. He/she can observe the motion of multiple pendulums that has different parameters and discover the laws in systems by comparing the motion. For example, setting only g_2 (the gravity

acceleration) to 1/2 of the default value in the right simulator, he/she can observe T_2 (the period$_2$ of motion of weight$_2$) becomes longer than T_1 (the period$_1$ of motion of weight$_1$). (3) Question and answer system is an implementation of a part of (3) ITS's features and can infer solutions and answer his/her questions by QSIM and DQ analysis. It can answer a question about the motion and its reason by QSIM and a question about the reason of change of motion by DQ analysis. However, it doesn't construct his/her comprehension state model by inferring his/her problem solving process, and support his/her comprehension by using the model.

4. Evaluation of Compound Comprehension Support System

We evaluated the compound comprehension support system in order to verify effectiveness of an assortment of multiple comprehension support methods. We prepared 6 questions. Testees were 12 students who belong to the faculty or the graduate school of information science. We divided them into two groups (averages of scores of the groups are equal in the pre test about dynamics). The testees took the following two tasks. Table 1 shows the rate of correct answer.

Task 1: In order to answer each question, the testees used a comprehension support method. The testees of group A used an unsuitable comprehension support method. The testees of group B used a suitable comprehension support method. The purpose is to verify strong and weak points of each comprehension support method.

Task 2: In order to answer each question, the testees used two comprehension support methods. The purpose is to verify effectiveness of an assortment of multiple comprehension support methods.

Table 1. Rate of Correct Answer

Group-Task	Comprehension Support Method(s)	Rate of Correct Answer
A-1	unsuitable	0.33
A-2	unsuitable + suitable	0.81
B-1	suitable	0.67
B-2	suitable + unsuitable	0.75

5. Conclusions

This paper proposed an effective comprehension support by an assortment of multiple comprehension support methods. In this study, we constructed and evaluated a compound comprehension support system about pendulums by assorting multiple comprehension support methods. We verified strong and weak points of each comprehension support method and effectiveness of an assortment of multiple comprehension support methods.

References

[1] Riichiro Mizoguchi, "Intelligent Tutoring Systems -the Current State of the Art-," The Transactions of the IEICE, Vol. E73, No. 3, pp. 297-307, 1990

[2] Toshio Okamoto, "The Current Situations and Future Directions of Intelligent CAI Research/Development," IEICE Transactions on Information & Systems, Vol. E77-D, No.1, pp. 9-18, 1994

[3] Yoshiki Kawaguchi, Hiromitsu Mori, Manabu Nakamura, and Setsuko Otsuki, "A Study on Assistance in Acquiring Meta-Cognition through Assorting Support Methods for Comprehension," Proc. of ICCE, Vol. I, pp. 104-105, 2002

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Applications of Data Mining in Constraint-based Intelligent Tutoring Systems

Karthik NILAKANT and Antonija MITROVIC
Intelligent Computer Tutoring Group, Department of Computer Science
University of Canterbury, Christchurch, New Zealand

Abstract. The number of ITSs being used daily is growing steadily. Consequently, huge amounts of interaction data are available, but data analysis is still very laborious. This paper describes the use of data mining processes to investigate student interaction with a constraint-based tutor. We discuss how statistical analyses, information visualization and machine learning algorithms can be used to discover interesting patterns in data, and how the findings can be used to improve the system.

1. Introduction

Several recent papers report results of data mining applied on ITS log files. AHA! [7] was mined using machine learning algorithms to find usage patterns. The evaluations of Logic-ITA [3] were able to identify common errors and the kinds of students who were likely to experience difficulty with the system. The data mining architecture for the LISTEN tutor is described in [2]. Mostow [6] highlights some important design aspects for data mining.

We performed a project involving SQL-Tutor, a constraint-based tutor that teaches the SQL query language. For details of system's implementation please see [4,5]. SQL-Tutor evaluates the student's solution, and generates feedback. After the first attempt, the student is only told whether the solution is correct or not (level 0 feedback). For subsequent submissions, the system increases the feedback level and provides more detail. For the second submission, the student would receive an *error flag* message, which specifies the part of the solution which is erroneous. Next, the student receives a feedback message originating from one of the violated constraints (level 2, *hint*). Higher levels of feedback are only available on request, and include *All errors* (level 3, providing feedback messages for all violated constraints), *partial solution* (level 4) and *complete solution* (level 5).

Nine evaluation studies were done with SQL-Tutor since 1998, and we have thus collected a lot of data which is hard to analyze. This paper reports on various types of data mining analyses we performed on the data set from the 2002 study. We approached the process of mining student logs as consisting of several phases. The first phase is *data collection*, and it produces raw data for later analyses. The second phase, *data transformation*, consists of converting data into appropriate formats and applying filter and aggregation techniques to raw data. The final stage, *data analysis*, is concerned with extracting interesting patterns from data, using statistical analysis, machine learning and information visualization. The data collection phase is already automated in our tutors. In the data transformation stage, the data was imported into a relational database, which allowed basic statistics to be gathered by querying the database.

2. Analyzing system logs

The distribution of attempts can be used as the basis for analysing the apparent difficulty of each problem. Table 1 ranks the top five problems using this measure. A human expert assigns a difficulty level (*Problem Complexity* in Table 1), ranging from 1 to 9 (the hardest problems), to each problem in SQL-Tutor. It is interesting to see that only one of the highest ranked problems has the complexity of 7, while the others are not viewed as being difficult. It is also interesting to see how many problems have been solved successfully. Of all attempted problems, 164 were solved by 5.6 students. The average complexity level for solved problems was 4.2, and this set involved problems at all levels. At the same time, 15 problems with the average complexity of 6.6 (attempted by 2.5 students) were never completed. These problems mostly involved aggregate functions and nested queries, which students find difficult to learn.

Table 1. Problem difficulty, based on average attempts per problem

Rank	Problem number	Problem complexity	Attempts per student
1	67	4	24.5
2	174	4	14.3
3	57	7	11.4
4	137	6	10.2
5	65	5	10

The analysis of constraint violations could help to identify difficult parts of the course. 447 constraints were relevant for submitted solutions, and were satisfied 87.5% of the time. Of these, 14 constraints (used on average in 14.6 attempts by 3.9 students) were never satisfied. These constraints are related to aggregate functions, joins and restrictions on grouping. From the remaining constraints, 156 were always used correctly.

We used the Apriori algorithm [1] to discover constraints that are frequently violated simultaneously. Table 2 shows some frequent sets of five constraints (identified by their numbers) that were commonly violated together. Constraint number 20, which was violated in 5.1% of all attempts (ranked 11th), and constraint 142 (ranked 15th) also occur together in a number of groups of constraints. The feedback statement given to the student after a violation of constraint 20 is "When you compare the value of an attribute to a constant in WHERE, they must be of the same type." Other often-violated constraints, such as 7, 142 and 147, are violated when parts of the student's solution reference database objects that do not exist. Constraint 239 is violated when a search condition is missing. This seems to indicate that students often use non-existent attributes in their WHERE clauses. Instead of generating a single feedback statement to alert students to this fact, the system generates a range of feedback messages that might confuse the student. For example, a feedback statement about a missing search condition may not be appropriate when they have been told the statement references a non-existent attribute. This analysis could be used as the starting point for redesigning constraints. For instance, groups of constraints that commonly occur together may be candidates for aggregation into a single, higher-level constraint. On the other hand, single constraints that tend to be violated more frequently than others may be candidates for decomposition into two or more lower-level constraints.

Table 2. Groups of constraints from SQL-Tutor

Most frequent sets with five items:
▪ 20 372 239 7 147 (64 instances) ▪ 142 372 239 7 147 (72) ▪ 142 20 239 7 147 (101)

The effectiveness of feedback could also be used as a topic for analysis. Among the topics of interest in this area are the distribution of different levels of feedback; deviations from the default feedback path; feedback patterns in groups of problems with varying complexity; and improvements (in constraint violations) resulting from feedback. One of the association rules the Apriori algorithm generated is:

(attempt_num = 0) [25%] → (feedback_level = 0) [88%]

This rule states that on first attempts (25% of the entire set), the likelihood of receiving level 0 feedback (the default feedback for the first attempt) is 88%. This means that only 12% of students explicitly change from the default feedback path on the first attempt. Table 3 shows the distribution of feedback levels, with respect to the attempt number. The default feedback path is mostly followed in the first three attempts. An interesting feature of the distribution is that by the 4th attempt, students begin to ask for the complete solution (level 5) more than on average. In the distribution of the second attempt, level 5 is the second most popular choice. Generally, if a student asks for the complete solution, it means that they have given up trying to solve the problem. It seems that around a fifth of all students give up after three attempts at a problem. Consequently, the ITS designer may consider not allowing the student to select this level of feedback until they make five attempts.

Table 3. Distribution of feedback levels, at different stages of problem solving

Attempt #	Instances	Level 0	Level 1	Level 2	Level 3	Level 4	Level 5
1	1544	88%	0%	2%	3%	0%	7%
2	992	9%	45%	14%	13%	4%	15%
3	821	6%	4%	44%	24%	5%	17%
4	645	5%	3%	37%	31%	4%	19%
5	458	7%	3%	38%	36%	4%	22%
>5	1454	10%	6%	21%	27%	7%	29%
Total	5914	29%	10%	20%	19%	4%	18%

3. Conclusions

The main advantage of data mining over traditional evaluation techniques is that information at a much finer level of granularity is available. Since this also means that more data are available for analysis, a more thorough investigation into student interaction is possible. This paper shows that diverse knowledge about system usage, problem difficulty, constraint violations and system feedback can be extracted from student logs.

One of the main areas for further research is to carry out similar studies with other constraint-based tutors. We are interested in applying data mining to data obtained from different student groups using the same tutor, students using different tutors, or using different versions of the same tutor.

References

1. Agrawal, R., Srikant, R. (1994) Fast algorithms for mining association rules. In: J. B. Bocca, M. Jarke and C. Zaniolo (eds), Proc. 20th Int. Conf. on Very Large Databases, Morgan Kaufmann, pp. 487-499.
2. Heiner, C., Beck, J., Mostow, J. (2004), Lessons on using ITS data to answer educational research questions. In: J. Mostow and P. Tedesco (eds) Proc. ITS 2004 Log Analysis Workshop, pp. 20–28.
3. Merceron, A., Yacef, K. (2003) A web-based tutoring tool with mining facilities to improve learning and teaching. In: U. Hoppe, F. Verdejo, J. Kay (eds) Proc. AIED 2003, IOS Press, pp. 201–208.
4. Mitrovic, A., Ohlsson, S. (1999) Evaluation of a constraint-based tutor for a database language. Int. J. Artificial Intelligence in Education, v10 no3-4, 238-256.
5. Mitrovic, A., Martin, B. and Mayo, M. (2002) Using Evaluation to Shape ITS Design: Results and Experiences with SQL-Tutor. *Int. J. User Modeling and User-Adapted Interaction*, v12no2-3, 243-279.
6. Mostow, J. (2004) Some useful design tactics for mining ITS data. In: J. Mostow and P. Tedesco (eds) Proc. ITS 2004 Log Analysis Workshop, pp. 20–28.
7. Romero, R. (2003) Discovering prediction rules in AHA! Courses. In: P. Brusilovsky, A. Corbett, F. de Rosis (eds) Proc. User Modeling 2003, Springer, pp. 25–34.

© 2005 The authors. All rights reserved.

Supporting Training on a Robotic Simulator using a Flexible Path Planner

Roger Nkambou[1] , Khaled Belghith[2], Froduald Kabanza[2], Mahie Khan[1]
[1] *Université du Québec à Montréal, Montréal (Québec) H3C 3P8, Canada*
nkambou.roger@uqam.ca ; khan.mahie@uqam.ca
[2] *Université de Sherbrooke, Sherbrooke (Québec) J1K 2R1, Canada*
khaled.belghith@USherbrooke.ca ; kabanza@USherbrooke.ca

Abstract. Manipulating the Space Station Remote Manipulator (SSRMS) on the International Space Station (ISS) is a very challenging task. The operator does not have a direct view of the scene of operation and must rely on cameras mounted on the manipulator and at strategic places of the environment where it operates. In this paper, we describe how a new approach for robot path planning called FADPRM can be used to support the training of astronauts on such a manipulator and under this big constraint of restricted sight.

Introduction

We are engaged in a research project for the development of an intelligent tutoring system called *Roman Tutor* to support astronauts in learning how to operate the SSRMS, an articulated robot arm mounted on the international space station (ISS). Astronauts operate the SSRMS through a robotic workstation located inside one of the ISS compartments. This workstation is equipped with an interface containing three monitors, each connected to one of the fourteen cameras placed at a strategic location on the ISS.

The SSRMS can be involved in various tasks that must be carried out very carefully on the ISS, ranging from moving a load to inspecting the ISS structure and making repairs. At different phases of a given manipulation, the astronaut must choose a setting of cameras that provides him with the best visibility while keeping a good appreciation of his evolution in the task. As most complex tasks deal in one way or another with moving the SSRMS, for the simulator to be able to understand students' operations in order to provide feedback, it must itself be aware of the space constraints and be able to move the arm by itself. A path-planner that calculates arm's moves without collision and consistent with best available cameras view is the key training resource on which other resources and abstract tutoring processes hinge. In this paper we describe how the FADPRM path-planner can be useful in providing amazingly tutoring feedback to a student (astronaut) during telemanipulation activities.

1. The FADPRM Path-Planner

In the literature, several approaches dealing with the path-planning problem for robots in constrained environments were found. Several implementations were carried out on the basis of these various approaches and much of them are relatively effective and precise. The fact is that none of these techniques deals with the problem of restricted sight we are faced with in our case. That's why we designed and implemented a new flexible and effective approach for robot path planning we call FADPRM [1].

FADPRM is a combination of the traditional PRM approach and AD*. It is flexible in that it takes into account zones with different degrees of desirability.

More specifically, FADPRM allows putting in the environment different zones with arbitrary geometrical forms. A degree of desirability *dd*, a real in [0 1], is assigned to each zone. On the ISS, the number, the form, the *dd* and the placement of zones reflect the disposition of the cameras on the station. The *dd* of a desired zone, such as a zone covering the field of vision of a camera, is then positive and the more it approaches 1, the more the zone is desired; the same for a nondesired zone where the *dd* is near 0. Thus, the FADPRM path-planner will try to bring the robot in zones offering the best possible visibility of the progression while trying to avoid zones with reduced visibility.

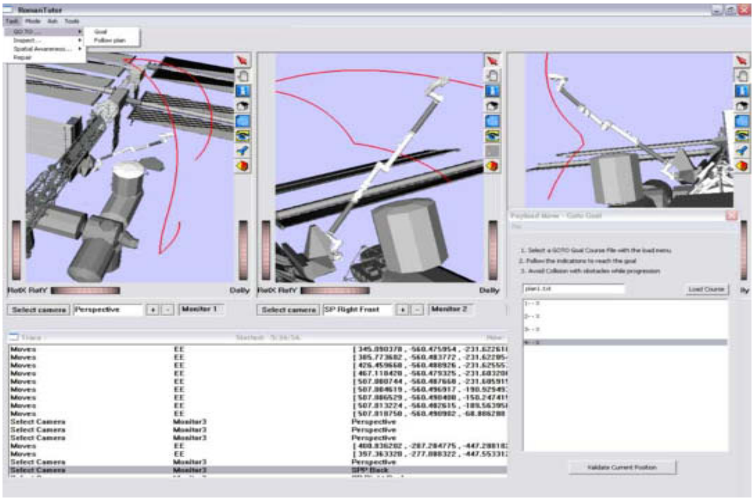
2. The Roman Tutor

The *Roman Tutor* user interface (Fig. 1) simulates that of the real workstation on which the astronauts operates the SSRMS. It contains three monitors with some buttons and functionalities to move the corresponding cameras: Tilt, Pan and Zoom.

The *Trace window* at the bottom of the *Roman Tutor* keeps a continuous track of all the operations done so far by the learner (the selection of a new camera in a monitor, the move of a camera and of the robot) and contains all information about the current state of the robot (if there is a collision or not, End-Effector's coordinates, cameras' positions etc.).

While the learner is carrying on some tasks on the simulator, traces of his progression are stored in files. The *Tutor* then parses these files in order to analyze the knowledge acquired by the learner, and to diagnose exactly where the gaps are. The *Tutor* diagnosis is done based on the plans generated by the FADPRM Planner and also on domain knowledge related to these plans. ISS and SSRMS domain knowledge is modeled as a Bayesian network which shows causal relationships between knowledge. *The Tutor* relies on this structure to diagnose students' errors in terms of lack of knowledge, misconception, etc.

The *Tutor* can also choose to provide the learner with a partial or total illustration of the task he's working on by calling the *Animation Generator*. The *Task Generator* allows the domain expert to design new tasks on which we might want to train astronauts.



```
Moves           EE          [ 345.090378 , -560.475954 , -231.622611
Moves           EE          [ 385.773682 , -560.483772 , -231.62205-
Moves           EE          [ 426.459668 , -560.488926 , -231.62555
Moves           EE          [ 467.118420 , -560.479325 , -231.683208
Moves           EE          [ 507.080744 , -560.487668 , -231.685911
Moves           EE          [ 507.084619 , -560.496917 , -190.525493
Moves           EE          [ 507.086529 , -560.490408 , -150.247411
Moves           EE          [ 507.013224 , -560.482615 , -109.563954
Moves           EE          [ 507.018750 , -560.490902 , -68.886288
Select Camera   Monitor3    Perspective
Select Camera   Monitor3    Perspective
Select Camera   Monitor3    Perspective
Moves           EE          [ 400.836202 , -287.284775 , -447.288183
Moves           EE          [ 397.363320 , -277.088322 , -447.553313
Select Camera   Monitor3    Perspective
Select Camera   Monitor3    SPP Back
```

1. Select a GOTO Goal Course File with the load menu
2. Follow the indications to reach the goal
3. Avoid Collision with obstacles while progression

plan1.txt

```
1--> 0
2--> 0
3--> 0
4--> 0
```

Figure 1. Roman Tutor User Interface (GoTo Task)

3. Using FADPRM Path-Planner for the Tutoring Assistance

One of the main goals of an intelligent tutoring system is to actively provide relevant feedback to the student in problem solving situations [2]. This kind of support becomes very difficult when an explicit representation of the training task is not available. This is the case in the SSRMS environment where the problem space associated with a given task consists of an infinite number of paths. *Roman Tutor* overcomes this problem by using FADPRM as principal resource for the tutoring feedback. *Roman Tutor* includes four different types of tasks: Spatial Awareness, GoTo, Inspect and Repair. The 'Spatial Awareness' task, improves the learner's knowledge about the space station's environment by providing him with some exercises such as naming and locating ISS elements, zones and cameras. In the 'GoTo' task (Fig. 1), the learner uses the SSRMS to move a load from one position to another different. The *Tutor* then executes the FADPRM planner, which generates a plan (a sequence of points) that joins the two positions. Based on this plan, the *Tutor* can: validate student action or sequence of actions, give information about the next relevant action or sequence, and generate relevant task demonstration resources.

During the evolution of the astronaut in a task, *Roman Tutor*, by calling the *Movie Generator* component, might choose to provide him with an animation illustrating entirely or partially the task he has to do. The movie generated after a call to the FADPRM path-planner, takes into account the disposition of the cameras on the station. In fact, it is constituted of a series of sequences taken from different and appropriate cameras showing the displacement of the SSRMS. For each sequence in the plan, the camera that gives the better sight of the displacement of SSRMS is chosen.

The learner is also provided with the 'Ask' menu, which allows him to ask different types of questions while executing a task. These questions may be of three different forms: How-To, What-If and Why-Not. *Roman Tutor* answers How-To questions by generating a path using FADPRM and by building an interactive animation that follow that path (as explained above). The incremental planning capability of FADPRM is used by *Roman Tutor* to bring answers to the What-If and Why-Not questions. In both cases, *Roman Tutor* provides the learner with relevant explanations given that his action or sequence of actions is out of scope of the generated plan or may bring him in a dead end.

4. Conclusion

In this paper, we described how a new approach for robot path planning called FADPRM could play an important role in providing tutoring feedbacks to the learner during training on a robot simulator. FADPRM is integrated into *Roman Tutor* allowing it to provide the learner with a continuous and relevant assistance during his progression in the task. Many tests and experiences have been carried out and we obtained very good and satisfactory results. In fact, we noticed that FADPRM enhances considerably the efficiency of the tutoring and improves the training quality. This constitutes a very important contribution in the field of intelligent tutoring systems. In fact, our results indicate that it is not necessary to explicitly create a complex problem space or task graph to support the tutoring process.

References

[1] F. Kabanza, R. Nkambou, K. Belghith, "*Path-Planning for Autonomous Training on Robot Manipulators in Space*". In Proc. of 19th International Joint Conference on Artificial Intelligence (IJCAI), 2005.

[2] K.VanLehn, "*The advantages of Explicity Representing Problem Spaces*". User Modeling 2003, Springer Verlag LNAI 2702:3.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

The eXtensible Tutor Architecture: A New Foundation for ITS

Goss NUZZO-JONES, Jason A. WALONOSKI, Neil T. HEFFERNAN, Tom LIVAK
Worcester Polytechnic Institute
100 Institute Rd, Worcester, MA 01609
(508) 831-5569
goss@wpi.edu, jwalon@wpi.edu, nth@wpi.edu, tomlivak@alum.wpi.edu

Abstract. The eXtensible Tutor Architecture (XTA) was designed as a platform for creating and deploying many types of Intelligent Tutoring Systems across many different platforms. The XTA presently has support for state graph pseudo-tutors and JESS model-tracing cognitive tutors, in both a client and server context. The XTA was designed with future development in mind, allowing easy specification of new tutor types, tutoring strategies, and interface layers. It has been used as the foundation of the Assistments Project, a wide scale web based ITS deployment. The Assistments Project is on track to provide ITS content to 100,000 students in the state of Massachusetts.

1. Introduction & Background

This research was conducted to develop a scalable, stable framework for deploying Intelligent Tutoring Systems (ITS) of many types to a variety of platforms, but was developed in particular based on the needs of the Assistments Project [3]. This project required that we be able to support a range of tutor types (constraint-based, cognitive-model, etc), provide stability and scalability, and deliver tutoring content to a host of clients – either rich client applications, or thin light-weight HTML clients. Additionally, we aimed to create an environment capable of supporting many tutoring strategies, operate as both a client and scaleable server application, provide logging capabilities for student analysis, and remain highly extensible for future development. To accomplish these goals, we employed component-based software engineering practices as well as judicious use of design patterns such as separation of logic and presentation. The results of this research were used as the deployment mechanism for the Assistments Project, a mathematics ITS project based at Worcester Polytechnic Institute and Carnegie Mellon University [3].

2. The eXtensible Tutor Architecture

The result of our research is a framework that we refer to as the eXtensible Tutor Architecture (XTA). This framework controls the interface and behaviors of our intelligent tutoring system via a collection of modular units. These units conceptually consist of a *curriculum* unit, a *problem* unit, a *strategy* unit, and a *logging* unit. Each conceptual unit has an abstract and extensible design allowing for evolving tutor types and content delivery methods. The current implementation has full functionality in a variety of useful contexts.

The *curriculum* unit represents a collection of educational content scheduled for tutoring. The *curriculum* is composed of one or more *sections*, with each *section* containing *problems* or other *sections*. This recursive structure allows for a rich hierarchy of different types of *sections* and *problems*.

The *section* component is an abstraction for a particular listing of problems. This abstraction has been extended to implement our current *section* types, and allows for future expansion of the *curriculum* unit. Currently existing *section* types include "Linear" (*problems* or sub-*sections* are presented in linear order), "Random" (*problems* or sub-*sections* are presented in a pseudo-random order), and "Experiment" (a single *problem* or sub-*section* is selected pseudo-randomly from a list, the others are ignored). Plans for future *sections* types include a "Directed" *section*, where *problem* selection is directed by the student's knowledge model [1].

The *problem* unit represents a problem to be tutored, including questions, answers, and relevant knowledge-components required to solve the problem. For instance, a problem could be implemented as a hierarchy of questions connected by correct and incorrect answers, along with hint messages and other feedback. Each of the questions represented by a *problem* composed of two main pieces: an *interface* and a *behavior.*

The *interface* definition is interpreted by the runtime and displayed for viewing and interaction to the user. To handle multiple target GUIs, the XTA deals with collections of low-level widgets with simple behaviors (such as text labels, fields, buttons, etc) that are translated into high-level widgets with complex behaviors (such as spell-checking text fields or algebra parsing text fields). This separation allows for easier conversion of interfaces towards a target UI (HTML or Swing, for example). A *behavior* definition for each *interface* is what allows each *problem* to define the results of actions (such as clicking a button) on the *interface.* Careful design keeps tutoring content developers focused on high-level widgets, while low-level representations can be ignored.

The *strategy* unit allows for high-level control over *problems* and provides flow control between *problems.* The *strategy* unit consists of *tutor strategies* and the *agenda*. Different *tutor strategies* can make a single *problem* behave in different fashions. Some types of *tutor strategies* that have already developed include message strategies (for hints, feedback, and instruction), explain strategies (for problem explanation), and forced scaffolding strategies (forcing a student into a particular branch of a problem). Future *tutor strategies* could include dynamic behavior based on knowledge tracing of the student log data, for example, which would allow for continually evolving content selection without a predetermined sequence of *problems*.

The final conceptual unit of the XTA is the *logging* unit, which receives detailed information from all the other units relating to all user actions and component interactions at every level of the system. Judicious logging can record the data required to replay or rerun a user's session to explore their misunderstandings of the content, reveal usage-patterns to aid in the detection of system gaming (superficially going through tutoring-content without actually trying to learn) [2], and provide useful reports to educators to enhance classroom instruction.

Each conceptual unit in the XTA is capable of being appropriately networked to leverage the benefits of distributing our framework over a network and across machines, to provide scalability. Also, based on memory footprint testing, thousands of copies of the XTA could run on a single machine. More importantly, the individual units described above are separated by network connections. This allows individual portions of the XTA to be deployed on different computers. Thus, in a server context, additional capacity can be added without software modification, and scalability is assured.

The XTA can also transform with little modification into a classic client-server architecture (such as Java WebStart) or a thin-client server configuration (HTML) depending on particular requirements. Both types of applications allow for pluggable client interfaces due to the *interface* definitions described earlier.

3. Methods and Results

The XTA has been deployed as the foundation of the Assistments Project [3]. This project provides mathematics tutors to Massachusetts students over the web and provides useful reports to teachers based on student performance and learning. The system has been in use for a year, and has had nearly 1000 total users. These users have resulted in over 1.3 million actions for analysis and student reports [2]. To date, we have had a live concurrency of approximately 50 users from Massachusetts schools. However, during load testing, the system was able to serve over 500 simulated clients from a single J2EE / database server combination. The primary server used in this test was a Pentium™ 4 with 1 GB of RAM running Gentoo Linux. Our objective is to support 3,000 users concurrently.

Tutors that have been deployed include scaffolding state diagram pseudo-tutors with a variety of strategies. We have also deployed a small number of JESS cognitive tutors for specialized applications. It should be noted that the tutors used in the scaling test described above were all pseudo-tutors, and it is estimated that a much smaller number of JESS tutors could be supported.

In summary, the launch of the XTA has been successful. The configuration being used in the Assistments project is a central server as described above, where each student uses a thin HTML client and data is logged centrally. Public school staff have enthusiastically reviewed the software. Since September 2004, the software has been in use at least three days a week over the web by a number of schools across central Massachusetts. This deployment is encouraging, as it demonstrates the stability and initial scalability of the XTA, and provides significant room to grow.

4. Conclusions

The larger objective of this research was to build a framework that could support 100,000 students using ITS software across the state of Massachusetts. We're encouraged by our initial results from the Assistments Project, which indicate that the XTA has graduated from conceptual framework into a usable platform (available at http://www.assistments.org). We have many planned improvements to the system including dynamic *curriculum* sections, *tutor strategies* that alter their behavior based on knowledge tracing of the student log data, and additional *interface* display applications for alternative GUI platforms. We believe the reconfigurable and customizable nature of the XTA could make it a valuable tool in the continued evolution of Intelligent Tutoring Systems.

References

[1] Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences*, 4 (2), 167-207.

[2] Feng, Mingyu, Heffernan, N.T. (2005). Informing Teachers Live about Student Learning: Reporting in the Assistment System. *Submitted as poster to the 12th Annual Conference on Artificial Intelligence in Education 2005, Amsterdam*

[3] Razzaq, L., Feng, M., Nuzzo-Jones, G., Heffernan, N.T., Koedinger, K. R., Junker, B., Ritter, S., Knight, A., Aniszczyk, C., Choksey, S., Livak, T., Mercado, E., Turner, T.E., Upalekar. R, Walonoski, J.A., Macasek. M.A., Rasmussen, K.P. (2005) The Assistment Project: Blending Assessment and Assisting. *The 12th Annual Conference on Artificial Intelligence in Education 2005, Amsterdam*

This research was made possible by the US Dept of Education, Institute of Education Science, "Effective Mathematics Education Research" program grant #R305K03140, the Office of Naval Research grant # N00014-03-1-0221, NSF CAREER award to Neil Heffernan, and the Spencer Foundation.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

An Agent-based Approach to Assisting Learners to Dynamically Adjust Learning Processes

Weidong PAN
Faculty of Information Technology
University of Technology, Sydney
PO Box 123 Broadway, NSW 2007, Australia

Abstract This paper presents an approach to assisting learners to dynamically adjust learning processes by using software agents. Software agents are integrated into the online learning environment to help learners develop personalized preferred learning plans and dynamically adjust learning towards their learning goals.

1. Introduction

Software agents have been applied in many different areas [1]. In this work, they are applied to assist learners to use a constructivist way to construct knowledge. Our agents assist learners not through understanding the academic content of subjects, but rather through providing a wide range of services that facilitate knowledge construction. One of the services is to assist them to dynamically adjust learning towards their goals [2].

According to the constructivist theories for learning, knowledge cannot be transmitted to learners but must be constructed by learners [3]. Researches have indicated that *not* all learners are equally capable of adequately constructing knowledge on their own [4]. Some may lack of necessary *prior* knowledge or abilities to independently choose appropriate learning resources and proper methods to conduct the process. Some may have no ideas of how to evaluate the outcomes of learning and vary plans according to real learning progress. Thus, it will significantly benefit learners to continue their pursuits for the goal if the instructional system can provide assistance for them to tackle these challenges.

Some intelligent tutoring systems help learners to solve these problems through learner models, expert models and tutorial models. They simulate learning processes and make all decisions for learners based on these models. The problem is that these models cannot possibly specify all of the ways in which learners may go about trying to solve a problem [5] due to their different backgrounds, interests, styles, motivations, capabilities, etc. It is even more true for online learning because most online learners are adult learners. As a result learners are often forced down the pre-set learning plans that do not suit them, or even limit the development of their cognitive abilities.

Our agents use a flexible approach to assist learners to address these challenges. Learners are not imposed to take any learning steps they do not like. Their autonomy in learning is supported and encouraged. Meanwhile the agents offer suggestions or advices for directing them to develop personalized learning plans and vary learning plans while they have troubles with these things.

2. The overall system architecture

Software agents with a hybrid architecture are incorporated into the online learning environment. Any change taking place in the learning environment made by learners in learning is detected by the agents. The learning progress is evaluated through evaluating the detected events. The learner profiles are built and timely updated through collecting the detected events and inducing from them. The agents take the real learning scenarios and the learner styles as input and generate suggestions or advices to facilitate productive learning.

In order for the agents to provide services tailored to an individual learner's *just-in-time* needs or even *just-for-me* needs, knowledge about the learning activities being conducted, the practical learning progress and the learner's learning styles is necessary. A UOL (unit of learning) database is being used to determine how to provide services based on this information. A UOL is a learning unit that satisfies one or more learning objectives. It may correspond to a course, a module, or even a single learning activity such as a discussion to elaborate on some topic. The UOL database is built by a series of carefully designed learning scenarios where not only the learning content and assessment methods are specified but also the relevant learning activities, the conduct sequences and the support services for various types of learners are defined as well. The structure of the UOL database is designed based on a UOL specification language which is being developed by us through extending and adjusting the EML [6].

3. The approaches to assisting learners to adjust learning processes

3.1 Assisting learners to develop personalized learning plans

The plan agent in the multi-agent architecture is responsible for assisting individual learners to develop a personalized plan to reach their learning goal. The assistance is implemented through advising them several plans for achieving their goals. These are the arrangements of the learning activities for the goals, extracted from the UOL database based on individual learning styles. The agent first determines a UOL by matching an individual learner's goal to the objective of a UOL in the UOL database. Then it collects all the possible learning methods for the UOL in the database. Next it further examines them and determines the ones which are suitable for the learner according to the fit degree of a method for him. Finally it presents these methods, i.e. the arrangements of relevant learning activities and their conduct sequences, as the recommended learning plans for the learner to adopt.

The agent uses the following technique to recognize if a particular learning arrangement is suitable for a learner and to measure the fit degree. The learning property of a learner, stored at his cognitive profile, is characterised by a set $P=\{p_1, p_2, \ldots, p_n\}$, where p_i is one of his preferred styles in learning, e.g. like to work together with others, like to work through concrete experiences, like to work with visualising images, etc. Every arrangement for a UOL stored in the UOL database has a similar set $M= \{m_1, m_2, \ldots, m_m\}$ describing its traits, where m_i is a style it can accommodate. The agent recognizes if an arrangement is suitable for a learner through comparing its M set against his preferred style set P. The fit degree is measured by summing the numbers where the learner's favoured styles are met by an arrangement, namely $V_{fit}=\sum (p_i \text{ in } M)$. An arrangement is recognized as an appropriate one for a learner if its V_{fit} is larger than a designated threshold value. An arrangement is considered as the optimal one if it has a larger V_{fit} than others.

3.2 Guiding learners to dynamically revise learning plans

Guiding learners to adjust learning towards their goals is implemented through managing individual learning plans. The UOL agents in the multi-agent architecture perform the work with the help of two lists, *activity list* and *check list.* Every learner is associated with an activity list and a check list for managing the progress of his learning plans. While a learner starts to learn a UOL, the UOL, the learning goal (i.e. goal UOL), and the adopted learning plan for the goal UOL are together put into his activity list. While a learner has completed the learning for a UOL, the UOL will be put into his check list. The activity list and check list are dynamical updated by the UOL agents.

The requirement for learning adjustment recognized by the agents takes place mainly in two situations. The first one is while a learner starts to learn a new UOL but he has not completed all the UOLs planned to learn *prior* to the one he is going to study. The agent first captures the learning plan the learner is adopting through a lookup table in his activity list based on the UOL he is going to study and the goal UOL. It then compares the check list for him against the learning tasks scheduled in the plan to see if the check list contains all the UOLs planned to learn *prior* to the one he is going to study. If it does not, the agent will suggest him adjust the learning by either revising the current plan or learning another UOL first. The second situation is while a UOL agent recognizes that he is not able to achieve the objectives of a UOL he is learning under the conditions at that time. An obvious case is he has failed to submit the desired artefact file for a long time after he started to learn a unit. Another case is the evaluation to his submission shows that he has not achieved the objectives of the UOL.

The suggestions offered by the agents not only contain a prompt suggesting a requirement for learning adjustment but also include the advice on how to align learning. In general, the possible way to adjusting learning can be two kinds: 1) to keep the learning plans being carried out unchanged and select another UOL to learn; and 2) to revise one of the plans being carried out. The agents generate a suggestion for the first kind of adjustments through comparing the current plan against the check list for the learner. It needs a complicated procedure to generate a suggestion for the second kind of adjustments. They will check all the possible plans for the current UOL and if no suitable one can be found, they will further check all the possible plans for the higher level UOL.

4. The perspectives

The paper presented an agent-based approach to assisting individual learners to develop personalized preferred learning plans and dynamically adjust learning based on the real progress of learning. The reported work is currently under implementation. We will be pursuing more personalized support from agents to assist individual learners to manage and adjust learning towards their goals. Meanwhile we will extend and refine the agent services in both scope and depth so that they will effectively assist online learners to construct knowledge.

References

[1] Jennings, N, Sycara, K. & Wooldridge, M. A Roadmap of Agent Research and Development. *Autonomous Agents and Multi-Agent System 1*. pp.7-38, 1998.

[2] Pan, W. & Hawryszkiewycz, I. To Develop Constructivist Learning Environments on the Web Using Software Agent Technology. *IASTED International Conference on Computer and Advanced Technology in Education*, Hawaii, USA, August 16-18. pp. 236-241, 2004.

[3] Jonassen, D. Constructivist Learning Environments on the Web: Engaging Students in Meaningful Learning. *Educational Technology Conference (EdTech 99).* Singapore, Feb 9-11, 1999.

[4] Large, A. Hypertext instructional programs and learner control: A research review. *Education for Information*, 14, pp. 95-105, 1996.

[5] Jonassen, D. *Computer as Mindtools for Schools: Engaging critical thinking*. Columbus, OH: Prentice-Hall, 2000.

[6] Koper, R. *Modeling units of study from a pedagogical perspective: the pedagogical model behind EML*. Available at: http://eml.ou.nl (10/12/2004)

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

EarthTutor: A Multi-Layered Approach to ITS Authoring

Kristen PARTON, Aaron BELL, and Sowmya RAMACHANDRAN
Stottler Henke Associates, Inc., San Mateo, CA
{parton, bell, sowmya}@shai.com

Abstract. This paper presents the EarthTutor authoring tool, a multi-layered system designed to remove the technical hurdles preventing teachers from having full control over the structure of their lessons, without sacrificing the power and technical flexibility required for an effective ITS implementation. EarthTutor reduces the ITS design problem to a series of related but independent layers. Each layer in the process hides the implementation details of lower layers, while providing a foundation for higher levels. This approach allows a core group of advanced authors to do the more complex authoring tasks and export their work to a wider audience of novice authors, who complete the authoring process.

1. Introduction

One of the primary goals of creating an authoring tool for Intelligent Tutoring Systems (ITSs) is to allow subject matter experts (SMEs), usually non-programmers, to easily build tutoring content. However, in [4], Murray argues that ITS design is by nature a difficult task, and requires “an adequate understanding of both the representational structure and the design process [of the ITS].” Because designing ITSs involves creating multifaceted components and the complex interactions between them, authoring tools can end up being very complicated, difficult-to-learn applications. On the other hand, creating an easy-to-use authoring tool may unduly constrain the ITS design, resulting in shallow or overly simplistic ITSs with no room for customization by the author. The EarthTutor authoring tool attempts to be both powerful and user-friendly by splitting the authoring into multiple layers with different interfaces. Each layer can then be developed by a different group of authors, which shifts the burden of complexity onto a small, core group of advanced authors, while allowing a larger group of novice authors to access the power of the authoring tool through a simpler interface.

2 Authoring Tool Overview

The EarthTutor ITS was designed for NASA to teach remote sensing image processing, a domain in which students analyze satellite data using an image processing application. EarthTutor’s objective is to convince professors and teachers to switch from using pen-and-paper course materials in a computer lab to using an interactive, adaptive ITS integrated with software the students are already using. Full integration with the host application allows EarthTutor to closely monitor the student as he completes real-world tasks in the application, as well as add instructional annotations to the application. The student is presented with a series of cards, which contain interactive behaviors embedded in HTML pages. Behaviors consist of questions and real-world tasks the student must complete in the host application. Authors specify the logic behind the tutoring behaviors, which includes monitoring the student's actions, presenting feedback, and updating the student model.

Figure 1 shows how components created in the EarthTutor authoring tool are used to build the student interface. The EarthTutor ITS core contains the general ITS components, such as student modeling, instructional planning, and the student GUI. Since EarthTutor works inside of a host application, there is also a plug-in layer that contains host-specific functionality (for instance, getting the active satellite image).

Both the ITS core and the plug-in layer export primitive functions called *actions* and *predicates* that allow interactive behaviors to access the ITS and host application. The advanced author interface contains a graphical interface for defining behaviors as flow charts (described in detail in [3]). The author builds flow charts using actions, predicates and other flow charts. Flow charts are parameterized and reusable, so flow charts can build upon other flow charts. Unlike a Finite State Machine, each behavior can also have local variables. Advanced constructs such as creating polymorphic behaviors, exception handling, and behavior scheduling parameters are also available for finer control, and the flow chart toolkit comes with an interactive debugger for rapid prototyping and development. Flow charts are exported to the novice interface as a *behavior library.*

In the novice interface, the author defines a hierarchical course structure, with a course containing labs, and labs containing cards. The author can set properties for the courses, labs and cards such as prerequisites and student modeling parameters. Each card contains HTML content and *behavior templates*. Behavior templates allow the novice authors to invoke the behaviors designed by the advanced authors by supplying the necessary parameters. Adding a behavior template to a card indicates 1) that the flow chart linked to the template should be executed when the card is displayed, and 2) that the student interface should replace the template with a UI component (defined by the advanced author in the flow chart). Behavior templates allow novice authors to tailor tutoring behaviors to their own pedagogical needs using parameters, but the interface is reduced to WYSIWYG HTML and simple forms.

The end result is an ITS that is much more interactive than a typical web-based ITS, and therefore, better at helping and assessing the students. For example, to teach a student how to calibrate an image, a less interactive ITS would explain the steps involved, ask the student to calibrate the image, then ask a follow-up question to check that the image was properly calibrated. In EarthTutor, a tutoring behavior would: 1) ask the student to open an image 2) wait for the student to open a file, then check that the file is correct 3) ask the student to use the calibration function and 4) wait for the student to calibrate, then check

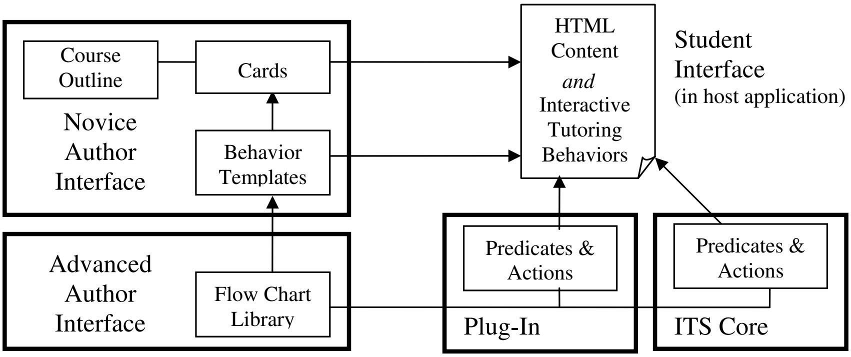


Figure 1. EarthTutor system architecture. The ITS core contains the student model, instructional planning, and displaying the student UI. The plug-in contains host-application specific functionality; e.g. allowing us to monitor the student's use of the image processing software.

the parameters of the calibration function. If the student uses the wrong parameters, the tutor will ask the student to undo the calibration and try again. Afterwards, the tutor can ask follow-up questions to make sure the student understood how the calibration works. Next time the student needs to calibrate, the tutor will simply ask him to calibrate an image, and expect him to remember the steps.

3 Discussion

The goal of the EarthTutor authoring tool was to provide content developers with control and flexibility, without sacrificing ease-of-use. Some authoring tools are designed to be easily used by classroom teachers, but either do not allow the author full control, or do not involve very much interactivity with the student. For instance, in REDEEM [8], interactions are limited to asking several types of questions, prompting the student to take notes, or directing the student's attention to another activity. REDEEM allows fine control over teaching strategy, but the teaching content is not very interactive. EarthTutor actually monitors the students as they complete real-world tasks, which enables the tutor to give more useful feedback.

In a multi-layer model, the inner layer can be thought of as a meta-authoring tool, which advanced authors use to create specialized, easy-to-use authoring tools for novice authors. As Murray notes in [3], this approach allows the authoring tool to "maintain depth, breadth *and* usability." Like the three-level model used by EASE [1], our model exploits the meta-authoring concept to provide power and usability. However, since the EarthTutor authoring tool allows users to switch between novice and advanced authoring interfaces, it is more flexible than a model which separates the meta-authoring level completely.

Separating the layers allowed the authoring work to be split between a small advanced group that designed behavior libraries, and a larger group of novice authors that used the flow charts. The behavior libraries also made it easier for SMEs to develop content, because the core Earth Science behaviors could be shared across the various labs and sub-domains (Oceans, Ozone, etc). Efficiency is a key factor in the authoring process, since creating an ITS is very labor intensive [4]. We believe that the multi-layer approach to ITS authoring increased authoring efficiency for EarthTutor, and detailed evaluations are planned in summer 2005 to get quantitative results on efficiency as well as usability.

Acknowledgements. This research is supported by NASA Small Business Innovation Research (SBIR) grant NNA04CA11C.

References

[1] Ainsworth, S. & P. Fleming: Teachers as Instructional Designers: Does Involving a Classroom Teacher in the Design of Computer-Based Learning Environments Improve Their Effectiveness? In Gerjets, P., Kirschner, P. A., Elen, J. & Joiner, R. (Eds.) (2004). *Instructional design for effective and enjoyable computer- supported learning. Proceedings of the first joint meeting of the EARLI SIGs Instructional Design and Learning and Instruction with Computers*, pp 283-291.

[2] Aroyo, L., Inaba, A., Soldatoba, L. & Mizoguchi, R.: EASE: Evolutional Authoring Support Environment, Proceedings of Intelligent Tutoring Systems (2004), 140-9.

[3] Fu, D., Houlette, R., Jensen, R. & Bascara, O.: A Visual, Object-Oriented Approach to Simulation Behavior Authoring, Proceedings of the Industry/Interservice, Training, Simulation & Education Conference (I/ITSEC) 2003.

[4] Murray, T.: Authoring Intelligent Tutoring Systems: An Analysis of the State of the Art, International Journal of AIED (1999), 10(1), 98-129.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Using Schema Analysis for Feedback in Authoring Tools for Learning Environments

Harrie PASSIER & Johan JEURING
Faculty of Computer Science, Open University of the Netherlands
Valkenburgerweg 177, 6419 AT Heerlen, The Netherlands
Email: harrie.passier@ou.nl & johan.jeuring@ou.nl

Abstract. Course material for electronic learning environments is often structured using schema languages. During the specification and development of course material, many mistakes can be made. We introduce schema-analysis as a technique to analyse structured documents, and to point out mistakes introduced by an author. With this technique we are able to produce valuable feedback.

Introduction

Electronic learning environments (LE's) are complex tools. Non-computer experts often write courses using such tools. Authoring tools have been developed to support the development of LE's. To improve the quality of LE's, an authoring tool should include mechanisms for checking the authored information on for example accuracy and consistency. Murray [7] mentions several such mechanisms. In this article we introduce schema-analysis as a technique to analyse course structure and domain ontology, which we represent by the languages IMS Learning Design (IMS LD) [5] and RDF. Using these flexible languages an author can easily make mistakes, which can be partly prevented by using templates. Some drawbacks of templates are loss of flexibility and problems with maintainability. With schema-analysis we maintain flexibility, are able to produce feedback when an author makes a mistake, and leave the author, as a didactic professional, free to accept or not accept the feedback information [3]. To show the technique at work we have defined and implemented six analyses to determine some quality aspects of a course: completeness, timely, recursiveness, correctness, synonyms and homonyms.

In this paper we briefly: explain what we mean with schemata, introduce the languages we use to represent them, describe in functional terms the analysis functions, and discuss some related work. A full paper and the (Haskell) code can be obtained from: http://www.ou.nl/info-alg-inf/Medewerkers/en_Passier.htm. The results presented in this paper are part of a project in which we investigate general feedback mechanisms to learners as well as to authors [8].

1. Schemata and representations

An ontology specifies the objects in a domain of interest together with their characteristics in terms of attributes, roles and relations. Using an ontology many aspects of a certain domain can be represented, for example categories and composition [9]. A composite object contains objects related to other objects using ‘ has_part’ or ‘uses’ relations and has structure. Such a structure description is called a script or a *schema*. In this article we focus on schemata. We use RDF to represent a domain ontology. The basic building block of RDF is a triple: <resource, property, value>, which defined concepts and related concepts. We use IMS LD [5] to represent the structure of a course. In this paper we focus on the Activity-model of IMS LD. To be able to add more specific annotations to content and structure we introduce two new elements in the Extra-p element: *Definition* and *Example*. Furthermore, we introduce a new attribute *Educational-strategy* of the element Activity with two possible values: *Inductive* (definitions after examples) and *Deductive* (examples after definitions). Introducing such elements will make it possible to structurally analyse educational material. Listing 1 shows some relevant elements related to the activity-model together with the newly defined elements example and definition. The new elements and attribute are marked in bold.

```
<!ELEMENT Activity  %Activity-model; >
<!ATTLIST Activity
          …
          Educational-strategy (Inductive | Deductive) >
<!ENTITY %Activity-model "(Metadata?, …, Activity-description)" >
<!ELEMENT Activity-description  (Introduction?, What, How?, …, Feedback-description?) >
<!ELEMENT What  %Extra-p; >
<!ENTITY %Extra-p "(…| Figure | Audio | Emphasis | List | … | Example | Definition)*" >
```

Listing 1. Parts of the activity-model in IMS LD definition

The elements *example* and *definition* include a short description, the central concept and the related concepts.

2. Schema analysis to detect authoring problems

We perform two types of analyses: 1) the analysis of structural properties of a schema, for example the recursive property, and 2) the comparison of a schema with one or more other schemata, for example to test the correctness of a definition in a course against an ontology.

We have developed six analysis functions that can be used to signal possible mistakes:

Completeness – We distinguish three kinds of (in)completeness: (1) within a course, (2) within an domain ontology and (3) between a course and an domain ontology. If a concept is used in a course, for example in a definition or an example, it has to be defined elsewhere in the course. A course is complete if all concepts used appear in the set of defined concepts. Completeness can also be applied to an (domain) ontology, and between a course and an ontology. The first one checks whether all used concepts in the ontology are defined in the same ontology, the second one if all used concepts in a course are defined in the ontology.

Timely – A concept can be used before it is defined. This might not be an error if the author uses an inductive instead of a deductive strategy to teaching, but issuing a warning is probably helpful. Furthermore, there may be a large distance (measured for example in number of pages, characters or concepts) between the definition and the use of the concept, which is probably an error.

Recursive concepts – A concept can be defined in terms of itself. Recursive concepts are often not desirable. If a concept is recursive, there should be a base case that is not recursive. Recursive concepts may occur in a course as well as in an ontology.

Synonyms – Concepts with different names may have exactly the same definition. For example, concept *a*, with concept definition (*a*, [*c,d*]), and concept *b*, with concept definition (*b*, [*c,d*]), are synonyms.
Homonyms – A concept may have multiple, different definitions. If for example concept *a* has definitions (*a*, [*b,c*]) and (*a*, [*d,f*]), then these two definitions are homonyms.
Correctness – The concepts in a course should correspond to the same concepts in its domain ontology.

The implementation of these analysis techniques are based on mathematical results about fixed points[4].

3. Related work

Although many authors underline the necessity of feedback in authoring systems [1][3][7], we have found little literature about feedback and feedback generation in authoring systems.

Jin et al [6] describe an authoring system that uses ontology's enriched with axioms to produce feedback to an author. On the basis of the axioms the models developed can be verified. It is not clear how general the techniques are. Aroyo et al. [1][2][3] describe an ontology based authoring framework, which monitors the authoring process and prevents and solves inconsistencies and conflicting situations. They list five requirements for such environments. We think that our framework[8] satisfies these requirements and that schema analysis supports these requirements. Stojanovic et al present an approach for implementing eLearning scenarios using the semantic web technologies XML and RDF, and make use of ontology based descriptions of content, context and structure [10]. A high risk is observed that two authors express the same topic in different ways (homonyms). This problem is solved by integrating a domain lexicon in the ontology and defining mappings, expressed by the author itself, from terms of the domain vocabulary to their meaning defined by the ontology.

4. Conclusions

This paper discusses schema analysis as a general technique to analyse structural aspects of learning environment related material.

References

[1] Aroyo L., Dicheva, D., Authoring support in concept-based web information systems for educational applications, in Int. J. Cont. Engineering Education and Lifelong Learning, Vol. 14, No. 3, 2004.

[2] Aroyo L., Dicheva D., The new challenges for e-learning: The educational semantic web, Educational technology & Society, 7 (4), 59 – 69, 2004.

[3] Aroyo, L. Mizoguchi, R., Towards Evolutional authoring support systems, Journal of interactive learning research 15(4), 365-387, AACE, USA, 2004.

[4] Davey B., Priestly H, Introduction to lattices and order, 2^{e} edition, Cambridge University Press, 2001.

[5] IMS LD: http://www.imsglobal.org/learningdesign/index.cfm.

[6] Jin L., Chen W., Hayashi Y., Ikeda M., Mizoguchi R., An ontology-aware authoring tool, Artificial intelligence in Education, IOS Press, 1999

[7] Murray, T., Authoring intelligent tutoring systems: An analysis of the state of the art. International Journal of AI in education, 10, 98 - 129, 1999.

[8] Passier, H., Jeuring, J., 2004. Ontology based feedback generation in design-oriented e-learning systems, Proceedings of the IADIS International Conferencee-Society 2004, Avila, Spain.

[9] Russell, S, Norvig, P., Artificial intelligence, A modern approach, Prentice Hall Int. editions, 1995.

[10] Stojanovic, L. Staab, S., Studer R., eLearning based on the semantic web, in WebNet 2001 – World conference on the www and internet, Orlando, Florida, USA, 2001

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

The Task Sharing Framework for Collaboration and Meta-Collaboration

Darren PEARCE, Lucinda KERAWALLA, Rose LUCKIN, Nicola YUILL and Amanda HARRIS

Ideas Lab, Informatics, University of Sussex

Abstract.

This paper describes a novel and general framework for the sharing of collaborative tasks between multiple users. In contrast to the majority of existing software used in a collaborative context, software developed under this framework provides each user with their own identical yet independent copy of the task which, by default, only they themselves can manipulate. This represents a departure from traditional turn-taking and dual control collaborative interaction styles and is intended to reduce the domination of one user over others as well as to provide the space for effective collaboration. The visual representation of agreement and disagreement is particularly emphasised since this has the potential to constructively mediate the resolution of collaborative disputes, especially if agreement is required at various points during the task.

Under the framework, it is also possible to *emulate* traditional collaborative interaction styles and, more importantly, dynamically vary between them, thus manipulating the collaborative affordances of the user interface. This is likely to scaffold conversation between the users about how they are working together to complete the task; users would be collaborating about their own collaborative process. The framework therefore holds significant potential to scaffold not only collaboration but also *meta*-collaboration.

Keywords. task sharing framework, collaboration, meta-collaboration,

1. Introduction

When software is used collaboratively, the nature of the user interface is critical. Various studies have explored collaborative interaction styles such as turn-taking and dual control (e.g. [1,2,3]) but, despite their popularity, it is still possible (and common) for one user to dominate others. This is often detrimental to the quality of the collaborative process.

In order to address this problem, this paper describes the Task Sharing Framework (TSF) by way of an example task in Section 2. Section 3 discusses the rich collaborative potential afforded by the introduction of this generally-applicable framework into the research area.

2. An Example TSF Task

Research in the Riddles Project[1] has used tasks in which children collaborate to create pictures to illustrate different interpretations of text. As an example, consider the task of illustrating a deliberately vague sentence such as *the girl heard the cat* by placing a girl and a cat on a background consisting of a tree.[2] One child may think that the girl has climbed the tree in order to rescue the cat whereas the other may think that the girl is on the ground, calling to the cat in the tree.

Using this example, Figure 1 demonstrates two of the ways in which the project has explored novel interfaces for collaborative tasks. The first, Figure 1a, consists of three panels. The two lower panels display each child's elements in the child's own 'colour': white for child 1 and diagonal stripes for child 2. The children are only able to move the elements that are within their *own* panel; they are *not* able to move their partner's elements. The upper, shared panel shows the elements belonging to *both* the children and also provides a representation of agreement and disagreement by colouring agreed elements in a distinct colour. Thus the cat is coloured black since both children agree it should be in the tree. On the other hand, the children disagree about the location of the girl so the panel shows her in each location using the relevant child's colour.

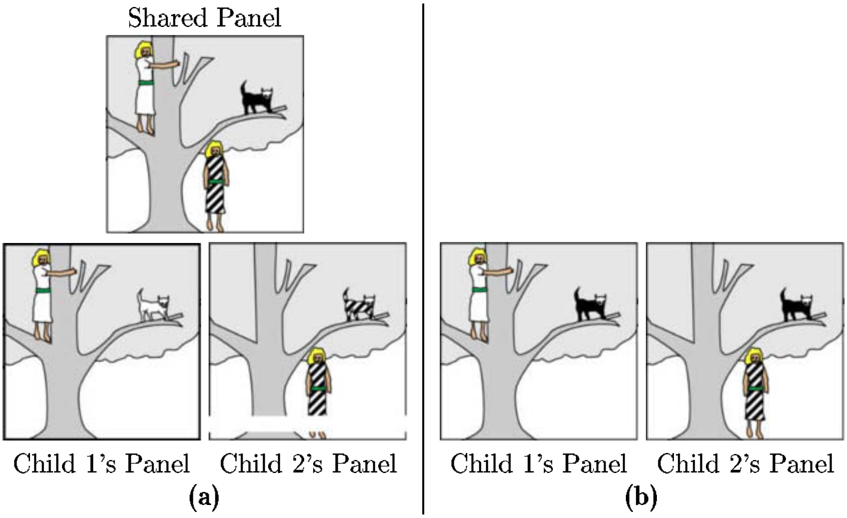


Figure 1. Two novel collaborative interfaces.

A second novel interface that the project has explored is shown in Figure 1b. In contrast to Figure 1a, it does not include a shared panel. Nonetheless, agreement and disagreement are still represented since each child panel *itself* highlights agreed elements. Thus in both child panels in Figure 1b, the cat is shown in black, the agreement colour.

[1] The work described in this paper was funded as part of the Riddles Project (`http://www.riddles.sussex.ac.uk/`), EPSRC grant number: GR/R96538/01.

[2] This task is based on [4].

3. Discussion

This example task demonstrates two of the key principles of the framework. Firstly, each user manipulates their own identical yet independent copy of the task. This is designed to reduce the potential for domination of one user over another and give them individual agency in the task process. Secondly, the framework emphasises the visual representation of agreement and disagreement. This has the potential to constructively mediate the resolution of collaborative disputes, especially if agreement is required at various points during the task.

TSF systems crucially have information about the task activity of each and every user. This carries significant implications for the scaffolding of collaboration since the system can effectively prompt users, highlighting differences between their current task states. The potential for this has been explored and shown to be effective [5]. It is also possible for a system to detect undesirable patterns of interaction such as one user consistently copying another. It could then intervene and prompt appropriately.

Under the framework, interaction styles are specified in terms of a common underlying representation. The flexibility of this representation makes it possible for TSF systems to *emulate* traditional interaction styles such as turn-taking and dual control. Furthermore, this flexibility means that a TSF system can *dynamically vary* between interaction styles potentially by using collected information on user interaction patterns. Such dynamic variation fundamentally changes the collaborative affordances of the user interface and is likely to lead to conversation between the users about how they are working together to complete the task. In this way, users would be collaborating about their own collaborative process. The framework therefore holds significant potential to scaffold not only collaboration but also *meta*-collaboration.

References

[1] K. Inkpen, J. McGrenere, K. S. Booth, and M. Klawe, "The effect of turn-taking protocols on children's learning in mouse-driven collaborative environments," in *Proceedings of Graphic Interface '97*, (Kelowna, BC), pp. 138–145, May 1997.

[2] S. Benford, B. Bederson, K.-P. Åkesson, V. Bayon, A. Druin, P. Hansson, J. Hourcade, R. Ingram, H. Neale, C. O'Malley, K. Simsarian, D. Stanton, Y. Sundblad, and G. Taxén, "Designing storytelling technologies to encourage collaboration between young children," in *Proceedings of the SIGCHI conference on human factors in computing systems*, (The Hague, The Netherlands), pp. 556–563, 2000.

[3] S. Scott, R. Mandryk, and K. Inkpen, "Understanding children's interactions in synchronous shared environments," in *Proceedings of Computer Supported Collaborative Learning (CSCL)'02*, (Bolder, CO, USA), pp. 333–341, January 2002.

[4] N. Yuill and T. Joscelyne, "Effects of organisational cues and strategies on good and poor comprehenders' story understanding," *Journal of Educational Psychology*, vol. 80, pp. 152–158, 1988.

[5] J. O'Connor and L. Kerawalla, "Using discussion prompts to scaffold parent-child collaboration around a computer-based activity," in *AIED 2005*, (Amsterdam), 2005.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Fostering Learning Communities based on Task Context

Niels PINKWART
University of Duisburg-Essen, Germany

Abstract. This paper presents an approach to establish and support learning communities. Based on task context information (which is extracted from multiple sources) and relying on the documents users created as primary source of information, the concept of a "peer recommender system" is presented which internally makes use of a mixture of different similarity measures – including, e.g., archive distance measurements and ontology based techniques.

1. Introduction

In recent years, the use of advanced computing and information processing techniques in education has significantly increased. Immediate benefits of digitally available resources come with the options of re-use and sharing. Yet, these are only some of the advantages that computational tools in education offer: other areas of great potential involve networked user communities. Mechanisms to intelligently interconnect learners and educational communities are a valuable goal, and can significantly contribute to advanced knowledge discovery and sharing. A reasonable starting point for this is to use the artifacts created by the users/learners, and to derive potential interaction partners based on this source of information. However, there are number of practical problems with this approach:

- The computational tools used within education are massively heterogeneous – even within content-level or educational domains. A common data format that could allow for generic access by analyzing algorithms is neither available nor realistic. Thus, the re-use of material across tools is not possible, and even the detection of interesting data is hard.
- Metadata that complies with accepted standards could solve at least the detection problem stated above. Yet, the process of manually indexing data is inconvenient and time consuming, so that people tend to avoid it, as there is usually little direct benefit [1].
- Mechanisms for determining potential interaction partners based on the content-level similarity of heterogeneous data are rare and not easy to define, especially under the criterion of simplicity in usage – which is important in educational settings.

The initial idea for this paper relies on the concept of *communication through artifacts*, a principle originally rooted in shared workspaces scenarios [2]. One problem with the concept is that, in its original form, it is restricted to synchronous cooperation with few participants in shared workspace scenarios. This paper presents an approach to retain the effects of artifacts used for mediating collaboration, while relaxing the constraints of time and group size. The approach relies on the following key ideas: (1) an exploitation of task context, which is provided by tools and archive systems, (2) the support of heterogeneous applications and data types through indirect mechanisms that allow the connection of users even though their preferred document formats might be incompatible, and (3) a mix of retrieval mechanisms which integrates, e.g., recommendation mechanisms and ontology based querying.

2. Existing approaches and technologies

A frequently chosen representation technique to address semantic interoperability concerns between data formats are *ontologies*. These can be understood as conceptualizations of a domain into a human-understandable, but machine-readable format consisting of entities, attributes, relationships, and axioms [3]. This explains why metadata systems often make use of ontologies as underlying structures – compared to other less structured and formal approaches, their suitability for AI and knowledge representation techniques can significantly increase or supplement classical information retrieval techniques.

Research on *recommender systems*, which aim at proposing relevant documents to users, is probably the most closely related area to the approach presented in this paper. Yet, their foundations differ from the approach presented in this paper in that recommender systems typically rely on (user provided) *ratings* of documents, which are either used directly for recommendation of the rated document, or indirectly to infer ratings for similar documents. A frequently applied method in the field of recommender systems is *collaborative filtering*, which has proven to be effective. Yet, it has several inherent problems, including the cold start problem. Several mechanisms to overcome this problem have been proposed. Some base on the idea of community membership, while others [4] investigate the synergies evolving from an integration of recommender systems with ontologies. Recently, also an approach which uses document assessments to dynamically update user profiles and build communities based on these profiles has been proposed [5].

3. Approach and System Architecture

The driving idea for the approach proposed in this paper differs from the listed concepts in that it does not require neither explicit nor implicit document assessment, but instead makes use of automatically available activity contexts for indexing and retrieval. A first feasibility study [6] illustrated how functionality required can even be embedded in the tool that provides the task context, thereby linking of user- and document-related metadata (e.g., including educational or domain specific dimensions, tool information, and user roles).

To flexibly interconnect heterogeneous tools with archives, a layered architecture can serve as a technical backbone for implementing the ideas presented in this paper. The facilities for the connection of users are then typically contained in a medium layer, consisting of four core components: (1) a *bridge* between the tools and the archives to allow for document uploads and downloads, as well as for transmitting queries and recommendations, (2) an *ontology* to store relevant domain/educational concepts and interrelations, (3) a representation of *task context* as extracted from archives, tools, and the ontology, and (4) an *AI based Community support engine* (CSE) that is in charge of calculating recommendations for potential interaction partners. The basic interface function of the CSE has the following signature:

```
recommend(doc)→[user_list]
```

The algorithm analyses the input document (content & context) and calculates a list of potentially interested users. `recommend` uses two functions to calculate the desired output:

- `related(doc)→[doc_list]` determines a set of documents that are similar to the parameter document, while
- `profile(doc,user)→[rating_list]` evaluates one document against all the documents created by a given user.

Both `profile` and `related` rely on a basic function which estimates the similarity of documents (including contexts):

$$\texttt{rate}(\texttt{doc}_1,\texttt{doc}_2) \rightarrow \texttt{rating}$$

This `rate` function builds the core of the CSE. It calculates a distance measure between documents with associated task contexts. However, in a realistic heterogeneous usage of the system concerning tools, data types, and probably a large number of missing or erroneous metadata pieces, good prediction qualities for the recommendation function are not easy to obtain, which motivates a more solid multi-method mix for the implementation of `rate`. This is addressed with the CSE structure illustrated in Figure 1.

- Even if the participating persons are new to the system and thus neither have a profile nor a document history, and the content data types are unknown to the system, the *ontology-driven engine* can estimate thematic proximity (in domain and educational terms) of data based on semantic context information.
- The *user profile comparison* takes general user/learner profiles (if available) as parameters, and allows for taking into account roles (e.g., teacher vs. student) or the languages of users. If available, also specific educational parameters of the user model can be taken into account here.
- Tool compatibility is an important information within the system, both directly (tool similarity calculated based on typology of tools), and also indirectly (data formats based, including compatibility information), as it allows for direct re-use of documents – this is addressed by the *tool-based similarity calculation.*
- The *archive distance measurement* uses inference techniques on an internal document/user graph to reveal document similarity of the type “most users who have seen document A have also looked at document B”.
- The *inner similarity check* allows tools to define content-level distance measures, and thus to incorporate domain-specific knowledge in the calculation process.

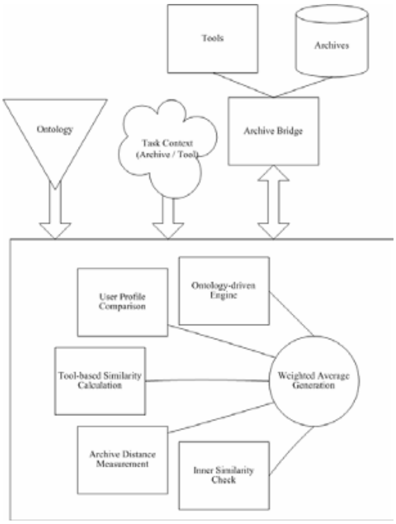

Tools
Archives
Ontology
Task Context (Archive / Tool)
Archive Bridge
User Profile Comparison
Ontology-driven Engine
Tool-based Similarity Calculation
Weighted Average Generation
Archive Distance Measurement
Inner Similarity Check

Figure 1. Architecture of the Community Support Engine

Finally, the *weighted average component* calculates a ranked user list based on the other single results – this list can then be used to recommend potential interaction partners.

References

[1] Wickens, C. D. (1992). *Engineering Psychology and Human Performance*. New York (NY), USA: Harper Collins.

[2] Dix, A.., Finlay, J., Abowd, G. D., & Beale, R. (2004). *Human-computer interaction*. Harlow, England: Pearson Education Limited.

[3] Guarino, N., & Giaretta, P. (1995). Ontologies and Knowledge Bases: towards a terminological clarification. In N. J. I. Mars (Ed.), *Towards very large knowledge bases: knowledge building and knowledge sharing*. Amsterdam: IOS Press.

[4] Middleton, S. E., Alani, H., & De Roure, D. C. (2002). Exploiting Synergy between Ontologies and Recommender Systems. In *Workshop Proceedings of WWW 2002*. Honolulu (HI), USA.

[5] Francq, P., & Delchambre, A. (2005). Using Document Assessment to Build Communities of Interest. In *Proceedings of SAINT 2005* (p. 327-333). IEEE Press.

[6] Pinkwart, N., Jansen, M., Oelinger, M., Korchounova, L., & Hoppe, H. U. (2004). Partial generation of contextualized metadata in a collaborative modeling environment. In L. Aroyo and C. Tasso (Eds.), *Workshop proceedings of AH 2004* (p. 372-376). Eindhoven (NL): Technical University Eindhoven.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

MALT - a Multi-lingual Adaptive Language Tutor

Matthias Scheutz [a] [1], Michael Heilman [a], Aaron Wenger [a], and Colleen Ryan-Scheutz [b]

[a] *Computer Science and Engineering Department, University of Notre Dame*
[b] *Department of Romance Languages and Literature, University of Notre Dame*

Abstract. We describe a "Multi-lingual Adaptive Language Tutor" (MALT) that uses natural language parsing and text generation to create various kinds of grammar exercises for learners of any language. These exercises can be restricted to specific topics by the instructor such as transformation of verb tenses. MALT generates novel exercises focusing on the specific difficulties of language learners as determined from their past mistakes, helping them overcome individual difficulties faster. We also present the first preliminary results from employing MALT in the foreign language classroom at Notre Dame.

Keywords. intelligent tutoring system, computer assisted language learning, Italian

1. Introduction

We propose a *Multi-lingual Adaptive Language Tutor* (MALT), which (1) addresses the problem of producing targeted individual exercises and producing feedback, and (2) has been successfully employed in a real language course at the University of Notre Dame. MALT consists of an automatic exercise generator that uses natural language processing methods to generate exercises dynamically and adaptively focus on the weaknesses of the language learner. MALT allows instructors to select a set of exercise types that focus on a particular grammar topic (e.g., transformation of verb tenses). MALT will initially present students with exercises generated from a random distribution of types from the set and record correct and incorrect answers. Based on the answers, it will quickly focus on "problem cases" and dynamically create appropriate exercises for them, thus providing learners with targeted exercises in areas, in which they are most likely to make mistakes. Hence, MALT avoids context-dependent learning effects where students learn only examplars, but not rules (such as learning the ending of a particular verb only in the context of a particular sentence with salient features). MALT provides detailed information to language instructors regarding the individual weaknesses and the learning trajectories of their students, which instructors can use as (one form of) feedback to complement their own written assessments and to help them adjust the content, pace, and sequence of materials in a given course. Finally, MALT can save both instructors and students time; the former by helping to produce different kinds of exercises for different students, the latter by focusing on problem cases without producing exercises on problem types, in which students have already demonstrated mastery. MALT currently has preliminary language modules for English, Italian, and Japanese.

2. A Brief Overview of the MALT System

MALT consists of *language-independent* modules (i.e., the syntactic parser, semantic representation and manipulation, text generation, question type selection, user interface) and *language-specific modules* (e.g., grammar rules, morphological rules including conjugation and declination tables, lexicon, etc.). This modular design allows for easy addition of new language modules as well as for adaptation of the user-interface to different teaching environments without having to change language-independent parts.

A top-down parser exercises are generated by based on initial non-terminal grammar symbols passed to the parser. The parser uses an *augmented context-free* grammar, where each terminal or non-terminal symbol may have one or more parameters attached to it. Augmented grammars allow for complex context-sensitive rules (e.g., as required for general formulations of subject-verb agreement) to be specified with relatively few grammar rules. Subject-verb agreement, for example, can be implemented by creating the following rule that specifies that the *number* parameter of the SUBJECT has to agree with that of the VERB: INTRANSITIVES→SUBJECT(NUMBER=?N)VERB(NUMBER=?N).

MALT targets question types according to past results stored in a simple *student model*. The model explicitly keeps track of the number of correctly and incorrectly answered questions for each category. For an exercise selection heuristic in these tests, types were selected based on which had been answered correctly the least number of times.

3. Experiments with MALT in Intermediate Advanced Italian

We conducted preliminary tests of MALT as part of the ROIT 215 *Intensive Intermediate Italian* course at Notre Dame in Spring 2005. Early in this course, students review the present tense subjunctive forms of regular and irregular Italian verbs for the three conjugations "-are", "-ere", and "-ire". These reviews typically include some more mechanical focus/practice for accuracy of forms with fill-in or transformation questions, where students are presented with a sentence containing the main verb in present tense indicative form, which they then have to transform into the present tense subjunctive form. We configured MALT to provide this kind of transformation exercise grouped into five question types (one for each conjugations, one for deviant "-ire", and one for irregular verbs). Moreover, MALT was embedded as an applet on a page in WebCT, a web-based teaching environment used in many classes at Notre Dame (but not in ROIT 215) [1].

We conducted a pre-test consisting of 20 transformation questions in WebCT before the in-class review of the material, after which students had one week to use the tutor voluntarily (instead of just practicing transformations using the workbook) before an in-class post-test again measured their performance.

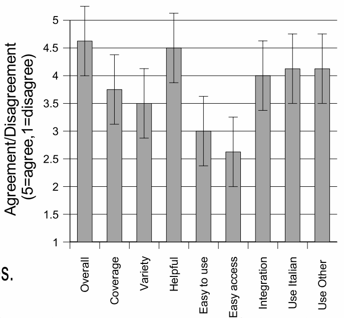
The results show that 6 out of the 7 students using MALT improved on the post-test (Post) based on the pre-test (Pre), while only 2 out of 5 students not using MALT showed improvement, indicating the utility of MALT as practice tool for verb transformations in

	without MALT					with MALT						
Pre	13	11	15	15	18	17	18	17	7	18	18	11
Post	11	20	17	14	15	14	19	18	14	19	20	16
Impr	0.85	1.82	1.13	0.93	0.83	0.82	1.06	1.06	2.00	1.06	1.11	1.45
Corr	-	-	-	-	-	19	57	79	177	60	170	20
Tot	-	-	-	-	-	24	65	115	267	67	185	30

Italian.[1] The lower part also shows the total number (Tot) and correct number (Corr) of practice transformations.

We also conducted an anonymous survey asking students to rate various categories, from “5–strongly agree” to “1-strongly disagree”:

SURVEY QUESTIONS:
MALT was overall useful.
MALT covered all conjugations and irregular verbs.
MALT produced a great variety of sentences.
MALT was helpful in practicing verb transformation.
MALT was easy to use.
MALT was easy to access.
MALT was well integrated into WebCT.
MALT should be used on a regular basis in Italian courses.



The overall results confirmed that students found MALT very useful and would like to see it integrated in Italian and other language courses.

4. Conclusion

The goal of MALT was to create an intelligent CALL system for realistic use in the foreign language classroom. Results from preliminary in-class tests of MALT with American undergraduate learners of Italian as a second language are very encouraging and justify a larger scale deployment, which is planned for Fall 2005 in the beginning Italian language courses. Future versions of MALT will provide a graphical interface that will allow instructors to customize MALT (e.g., by adding or deleting lexical items, grammatical rules, question types, etc.–currently this is only possible via text files and requires knowledge of specific data structures in MALT).

References

[1] WebCT, HTTP://WWW.WEBCT.COM/

[2] J. Gamper, J. Knapp. “A Review of Intelligent CALL Systems.” *Computer Assisted Language Learning* 2002 Vol. 15, No. 4 , pp. 329-342. Swets and ZeitLinger.

[1] A t-test comparing the average improvements *Impr* (given by the fraction Post/Pre) of the two groups (1.22 with and 1.11 without MALT) is not statistically significant ($p = 0.65$), which is partly due to one's student large improvement and also to the small sample size (we expect the difference to become significant with a larger group).

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Teaching the evolution of behavior with SuperDuperWalker

Lee Spector [a,1], Jon Klein [a,b], Kyle Harrington [a] and Raymond Coppinger [a]
[a] *Cognitive Science, Hampshire College, Amherst, MA, USA*
[b] *Physical Resource Theory, Chalmers U. of Tech. and Göteborg U., Göteborg, Sweden.*

Abstract. SuperDuperWalker is a software-based framework for experiments on the evolution of locomotion. It simulates the behavior of evolving agents in a 3D physical simulation environment and displays this behavior graphically in real time. A genetic algorithm controls the evolution of the agents. Students manipulate parameters with a graphical user interface and plot outputs using standard utilities. The software supports an inquiry cycle that has been piloted in CS193T: Biocomputational Developmental Ecology at Hampshire College.

Keywords. Physical simulation, genetic algorithms, biology,

The science curriculum at Hampshire College[2] emphasizes original student inquiry at all levels, including first-semester courses and courses for non-majors. This puts a premium on tools and methodologies that allow for genuine inquiry by novices. As a result, Hampshire College faculty have developed a range of methodologies and technologies for student-active inquiry-based science education, and they have also studied the efficacy of these methods [1,4,7,8,9].

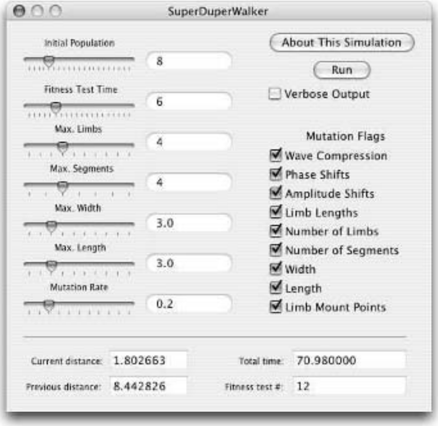
Klein's *breve* simulation system [3] allows programmers to quickly build interactive physical simulations that are rendered in realtime using the OpenGL 3D graphics library. It supports arbitrary computations expressed in an object-oriented language and it also supports the integration of customized graphical user interfaces. While developed primarily for experiments in complex adaptive systems and artificial life (e.g. [6]), it has also proven useful as an environment for inquiry-based courses on artificial life, artificial intelligence, and algorithmic arts.[3] In the present contribution these methods are applied to a course for first-semester students on issues in evolutionary biology.

The artificial intelligence technology in this system is used not to automate pedagogy (although such extensions are conceivable) but rather to produce a rich virtual world in which experiments can be conducted. The "biology" of this virtual world is, of course, an abstraction that differs in innumerable ways from that of the real world, but it nonetheless allows students to explore important principles of evolutionary dynamics.

[1] Correspondence to: Lee Spector, School of Cognitive Science, Hampshire College, Amherst, MA 01002-3359, USA. Tel.: 413-559-5352; Fax: 413-559-5438; E-mail: lspector@hampshire.edu. This material is based upon work supported by the USA NSF under Grant No. 0308540 and Grant No. 0216344, and by DARPA/AFRL/USAF, under agreement number F30502-00-2-0611. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation.

[2] See http://www.hampshire.edu

[3] See for example http://hampshire.edu/lspector/cs263/cs263s04.html



Initial Population: 8
Fitness Test Time: 6
Max. Limbs: 4
Max. Segments: 4
Max. Width: 3.0
Max. Length: 3.0
Mutation Rate: 0.2

□ Verbose Output

☑ Wave Compression
☑ Phase Shifts
☑ Amplitude Shifts
☑ Limb Lengths
☑ Number of Limbs
☑ Number of Segments
☑ Width
☑ Length
☑ Limb Mount Points

Current distance: 1.802663
Previous distance: 8.442826
Total time: 70.980000
Fitness test #: 12

Figure 1. The SuperDuperWalker graphical user interface.

Klein's "Walker" program, which is included as a demo in the *breve* distribution, uses a genetic algorithm in a manner inspired by Dawkins's "Biomorph" [2] and Sims's virtual creatures [5] to evolve four-legged walking creatures. Creatures that travel longer distances than their competitors are allowed to reproduce and to produce children that are varied by mutation. The "SuperWalker" program, which is also included as a demo in the *breve* distribution, adds an additional degree of freedom, allowing not only the leg controllers but also the leg segment lengths to evolve. SuperDuperWalker further extends SuperWalker by adding many more degrees of freedom (for example the number of legs, number of leg segments, leg placements, and several other parameters may also evolve) and by providing a graphical user interface (see Figure 1) that allows non-programmers to conduct experiments.

Evolving creatures are displayed in real time as they compete with one another in fitness tournaments. Snapshots of two individual creatures are shown in Figures 2 and 3. The movies from which these snapshots were taken are available online, along with SuperDuperWalker source code and related teaching materials.[4] The software produces tabular output that can be imported into off-the-shelf spreadsheet software, manipulated, and graphed (as in Figure 4).

The software was used in a fall semester, 2004 course at Hampshire College, CS193T: Biocomputational Developmental Ecology. The instructors demonstrated the software and its use in an inquiry cycle of hypothesis formation, experiment design, data collection, and analysis. Students were then expected to conduct their own inquiry cycles (based on their own hypotheses) in an in-class lab and to produce a lab report. Several students also used SuperDuperWalker experiments as the basis of their final projects at the end of the semester.

We expect the technology used in SuperDuperWalker, which combines physical simulation with a graphical user interface and the strategic use of artificial intelligence algorithms (such as genetic algorithms), to present additional opportunities for inquiry-based education across the curriculum.

[4] http://hampshire.edu/lspector/superduperwalker.html

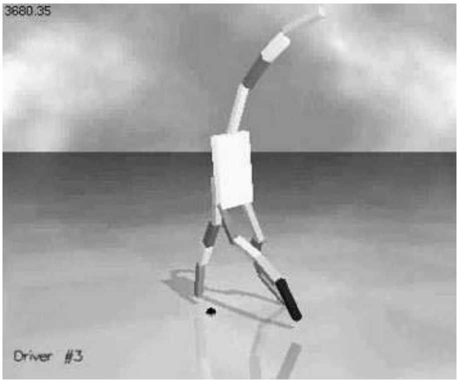


Figure 2. A snapshot of an evolved creature.

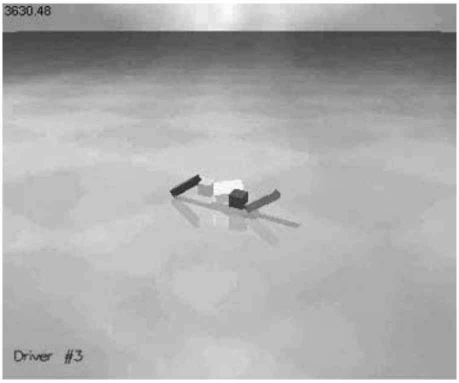


Figure 3. A snapshot of an evolved creature.

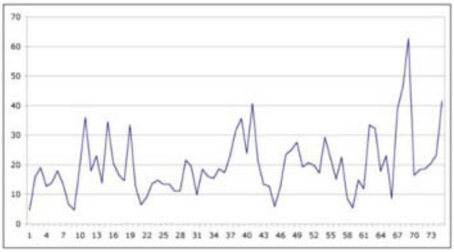


Figure 4. A graph of average distances traveled (on the y axis) for each 4-creature tournament (with later tournaments to the right) over the course of a SuperDuperWalker run.

References

[1] M. S. Bruno and J. N. Chase, eds., *School/College Partnerships: Inquiry-Based Science and Technology for All Students and Teachers*, Amherst, MA: Science & Technology Education Partnership at Hampshire College, http://demeter.hampshire.edu/ manual/, 1997.

[2] R. Dawkins, *The Blind Watchmaker: Why the Evidence of Evolution Reveals a Universe Without Design*, W. W. Norton & Co., 1996.

[3] J. Klein, breve: a 3D simulation environment for the simulation of decentralized systems and artificial life, in *Proceedings of Artificial Life VIII, the 8th International Conference on the Simulation and Synthesis of Living Systems*, The MIT Press, 2002.

[4] A. McNeal and C. D'Avanzo, *Student-active Science: Models of Innovation in College Science Teaching*, Saunders College Press, 1997.

[5] K. Sims, Evolving Virtual Creatures, in it Proceedings of SIGGRAPH, 1994, pp.15–22.

[6] L. Spector, J. Klein, C. Perry, and M. Feinstein, Emergence of Collective Behavior in Evolving Populations of Flying Agents. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 2003, Berlin: Springer-Verlag, pp. 61–73.

[7] N. Stillings and L. Spector, Inquiry-Based Learning: Cognitive Measures & Systems Support, http://hampshire.edu/lspector/NSF-LIS/, 1999.

[8] S. E. Weisler and Slavko Milekic, *Theory of Language*, MIT Press, 1999.

[9] S. Weisler, R. Bellin, L. Spector, and N. Stillings, An Inquiry-based Approach to E-learning: The CHAT Digital Learning Environment, in *Proc. SSGRR-2001, Intl. Conf. Adv. in Infrastructure for Electronic Business, Science, and Education on the Internet*, http://www.ssgrr.it/en/ssgrr2001/papers.htm, 2001.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Distributed Intelligent Learning Environment for Screening Mammography

Paul TAYLOR[*], Joao CAMPOS[*], Rob PROCTER[**], Mark HARTSWOOD[**], Louise WILKINSON[†], Elaine ANDERSON[††], Lesley SMART[††]
[*]*Centre for Health Informatics, University College London, UK*
[**] *School of Informatics, University of Edinburgh, UK*
[†] *St George's National Training Centre, London, UK*
[††]*South East Scotland Breast Screening Centre, Edinburgh, UK*

Abstract. We are developing a computer based training system to support breast cancer screening. The prototype we are developing is a *distributed* Intelligent Learning Environment in which the work of collecting and annotating interesting cases will be shared and the experience of using the images will be pooled. It is only by creating a network of students and mentors across different clinical sites that we can hope to obtain the breadth and variety of experience required to build robust models of the screening context. Using a pragmatic approach, our design allows for exploratory and experiential learning. Such a design is more likely to succeed in a screening environment because the system will fit the needs of film readers without being prescriptive about how and what they should learn

1. Introduction

1.1 Mammography and Screening Practice

Breast screening programmes invite women of a certain age group for regular assessment. They undergo an X-ray examination involving one or more views of each breast (called mammograms). These mammograms are, ideally, reviewed (independently) by at least two film readers. Readers compare current with previous films (where available) and identify mammographic features that may be indicative of cancer. Film readers are aware of their responsibility both to detect cancers and to avoid recalling healthy women unnecessarily.Although mammography provides an acceptable trade off between cost, sensitivity and specificity, it is known to be an imperfect screening technique [1].

1.2 Film Reading Expertise

Researchers have reported high variations in readers' performance [2]. It seems likely that the acquisition of expertise requires planned and targeted training of a standard that is often unobtainable in practice. Only about six cancers are found for every thousand women screened, a trainee who learns by working alongside an expert colleague could read films from 200 cases a week for six months and see as few as thirty cancers. Textbooks and mammography atlases are commonly used to support training, however, computer-based training systems (CBT) are expected to become an integral part of radiological training.

CBT systems will give their users access to a much greater variety of images and support learning in ways that can be fitted into a busy clinical schedule.

2. Designing a Computer Based Training System for Screening Mammography

The underlying learning goals (what the system will teach) and teaching strategies (the ways in which the system will teach) of CBTs are reflected in the way the information is presented in the course of a training session. Most attempts to provide such a tool for mammography have, however, been based on relatively modest image databases and offer a limited educational experience. We wish to explore research into applications of artificial intelligence in educational software, more specifically, the design architectures of intelligent tutoring systems (ITS) and intelligent learning environments (ILE).

ITSs are based on the cognitive theory of skill acquisition, and incorporate a number of instructional principles and methods from this theoretical framework. Such systems follow an objectivist view of the nature of knowledge. In contrast, ILEs follow a constructivist view, assuming that knowledge is individually constructed from what the learners do through interacting with an environment [3]. ILEs, therefore, contain knowledge about the context in which learning takes place and the activities in which the user is expected to engage, in order to provide a rich and flexible environment.

Our field work has revealed that the important decision in screening, whether or not to recall a woman for further tests, is based on an intuitive assessment of the risks associated with each atypical appearance and the appraisal of that assessment in the specific context of each individual screening centre. This can not modelled in the way that the objectivist approach would require. The constructivist approach, therefore, seems more appropriate for screening. Translating its concepts to computational terms, our system allows for exploratory and experiential learning in which the user chooses different ways of doing things, reflects on the actions taken and the system, based on observation of the user's actions, suggest alternative pathways. In this way, the system will fit the user without being prescriptive about what and how they learn.

We believe that an effective CBT for screening mammography can only be created through collaboration. It is only by creating a network of students and mentors across different clinical sites that we can hope to obtain the breadth and variety of experience required to build robust models of the screening context. We are developing a distributed ILE in which the work of collecting and annotating interesting cases will be shared and the experience of using the images will be pooled. The system will offer a number of benefits:

- *Availability* A digital archive of cases will give instant access to a wide range of training materials, and thereby open up new opportunities for training delivery.
- *Completeness* An ILE can broaden substantially the range of cancerous presentations to which a trainee is exposed.
- *Automation* An ILE can automate aspects of the training process, for example, the marking of a trainee's decisions and tracking performance.
- *Collaborative Working* An ILE allows: remote delivery of training supervised by mentors in accredited training centre; support for 'asynchronous' supervision in centres where mentors and trainees are co-located; the opportunity for trainees to share their experiences regardless of whether they are co-located.

- *Statistical analysis* Aggregate statistics of performance on lesions, cases and training sets over repeated application by trainees with differing levels of experience will be used to provide metrics of 'difficulty'. These assist with the allocation of training sets commensurate with a trainee's current expertise and with the compilation of new training sets to facilitate the learning of clinically important distinctions.

The tool is based around a screening workstation, so that the training takes place within an environment that supports routine clinical work. Users have access to a large database of selected cases, including a mix of typical and unusual cases, both normals and cancers. The prototype allows users to select a training roller from a menu of cases with different characteristics, to attempt interpretation case by case and receive feedback, both in the form of advice about the interesting features of each case and through statistics of their overall performance. The difficulty of the tasks may be adjusted. The system also suggests areas that the user might review or to concentrate on, and keeps a record of what the user has done. In this way, the training system can induce users to reflect on strategy and plans.

3. Discussion and Future Work

Our work is carried out as part of a larger project [4] set out to demonstrate the benefits of Grid technology for breast imaging in the UK. Over the last two years, team members have conducted lengthy observational studies of screening work and training sessions. Senior radiologists have been closely involved in the design. The environment is a good approximation to a high-quality digital screening workstation. The didactic information that the tool provides is based on careful and scrupulous annotation of a large database of images by experienced radiologists with an interest in training.

We show how a detailed understanding of screening work influences the design of a CBT tool. Some aspects of screening are embedded in a context and therefore hard to formalise. Using a pragmatic approach, we are now designing a system to allow for exploratory and experiential learning. Such a design is more likely to succeed in a screening environment because the system will fit the needs of film readers without being prescriptive about how and what they should learn. Our design will permit experiments to evaluate how users explore the available data; to collect data on user performance, skill and expertise; and on individual case difficulty and roller composition.

References

[1] Fitzgerald R. Error in Radiology. *Clinical Radiology* 2001; 56:938 – 946.

[2] Beam C, Layde M, Sullivan D. Variability in the interpretation of screening mammograms by US radiologists. Findings from a national sample. Archives of Internal Medicine 1996; 156: 209 – 213.

[3] Akhras, F. and Self, J. (2000) *System Intelligence in Constructivist Learning*. International Journal of Artificial Intelligence in Education, 11(4):344-376.

[4] Brady JM, Gavaghan DJ, Simpson AC, Parada MM, and Highnam RP. (2003) *eDiaMoND: A Grid-enabled federated database of annotated mammograms.* In Berman, Fox and Hey, *Grid Computing* p 923-943, Wiley.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

The Assistment Builder: A Rapid Development Tool for ITS

Terrence E. TURNER, Michael A. MACASEK, Goss NUZZO-JONES, Neil T. HEFFERNAN
Worcester Polytechnic Institute
100 Institute Rd, Worcester, MA 01609
(508) 831-5569
tinylox@wpi.edu, macasek@wpi.edu, goss@wpi.edu, nth@wpi.edu
Ken KOEDINGER
Human-Computer Interaction Institute
Carnegie Mellon University, Pittsburgh, PA, USA

Abstract. Intelligent Tutoring Systems are notoriously costly to construct [1], and require PhD level experience in cognitive science and rule based programming. The goal of this research was to ease the development process for building pseudo-tutors [5], which are ITS constructs that mimic cognitive tutors but are limited in that they only work for a single problem. The Assistment Builder is a system designed to rapidly develop, test, and deploy simple pseudo-tutors. These tutors provide a simple cognitive model based upon a state graph tailored to a specific problem. These tutors offer many of the features of rule-based tutors, but without the expensive creation time. The system simplifies the process of tutor construction to allow users with little or no ITS experience to develop content. The system provides a web-based interface as a means to build and store these simple tutors we have called *Assistments*. This paper describes our attempt to make the process of developing content easy for teachers. We present some evidence to suggest that these novice users can develop a tutor for a problem in under thirty minutes.

1.0 Introduction

This research aims to develop tools for the rapid development and deployment of Intelligent Tutoring Systems (ITS). Specifically, this research focused on so-called "pseudo-tutors" that are a simplification of cognitive rule-based tutors [5]. Model tracing rule-based tutors [1] have been shown to be effective [6], but development time on them is highly prohibitive, from 100-1000 hours of development time per hour of content [7][1]. Development also requires a very specialized knowledge set. Tutor developers are required to be expert system programmers, in addition to developing the cognitive model, to say nothing of being a content expert. Another aim of this research was to make our tools accessible to novices, with no programming experience, and less than an hour of training.

A pseudo-tutor is a simplified cognitive model based on a state graph. Student actions trigger transitions in the graph, and the current state of the problem is stored by the graph. Pseudo-tutors have nearly identical behavior to a rule-based tutor, but suffer from having no ability to generalize to different problems [4]. This pseudo-tutor approach allows for predicted behaviors and provides feedback based on those behaviors. We also combined this state graph with a conceptually broader branching structure referred to as scaffolding. Scaffolding provides sub-problems to the initial question, often designed to address specific concepts within the initial question. This allows for a higher-level of predicted actions to be handled.

1.1 Purpose of the Assistment Builder

The Assistment Builder is an application supporting the Assistment Project [8]. We sought to create a tool that would provide a simple web-based interface for creating these pseudo-tutors that could rapidly be deployed across the web, and if errors were found with the tutor, bug-fixing or correction would be quick and simple. The tool had to be usable by someone with no programming experience or ITS background. We wanted the teachers in the public school system to be able to build pseudo-tutors. These pseudo-tutors are often referred to as *Assistments*, but the term is not limited to pseudo-tutors.

A secondary purpose of the Assistment Builder was to aid the construction of a Transfer Model. A Transfer Model is a cognitive model construct divorced from specific tutors. The Transfer Model is a directed graph of *knowledge components* representing specific concepts that a student could learn. This allows us to maintain a complex cognitive model of the student without necessarily involving a production rule system.

When a user first begins to use the Assistment Builder they will be greeted by the standard blank skeleton question. The user can enter the question text, images, answers, and hint messages to complete the root question. After these steps the appropriate scaffolding is added. The question layout is separated into several views the *Main View*, *All Answer View*, *Correct Answer View*, *Incorrect Answer View*, *Hints View*, and *Transfer Model View*. Together these views allow a user to highly customize their question and its subsequent scaffolding.

2.0 Methods

To analyze the effectiveness of the Assistment Builder, we developed a system to log the actions of an author. Each action is recorded with associated meta-data, including author, timestamps, the specific series of problems being worked on, and data specific to each action. The authors were asked to build original items and keep track of roughly how much time spent on each item for corroboration. The authors were also asked to create "morphs," a term used to indicate a new problem that had a very similar setup to an existing problem. "Morphs" are usually constructed by loading the existing problem into the Assistment Builder, altering it, and saving it with a different name. This allows rapid content development for testing transfer between problems. We wanted to compare the development time for original items to that of "morphs" [8].

Another trial of the Assistment Builder with less rigorous methodology was testing how authors with little experience would react to the software. To test the usability of the Assistment Builder, we were able to provide the software to two high-school teachers in the Worcester, Massachusetts area. These teachers were computer literate, but had no previous experience with intelligent tutoring systems, or creating mathematics educational software. Our tutorial consisted of demonstrating the creation of a problem using the Assistment Builder, then allowing the teacher to create their own with an experienced observer to answer questions.

3.0 Results & Analysis

Prior to the implementation of logging within the Assistment Builder, we obtained encouraging anecdotal results of the software's use. A high-school mathematics teacher was able to create 15 items and morph each one, resulting in 30 *Assistments* over several months. Her training consisted of approximately four hours spread over two days in which she created 5 original *Assistments* under supervision. While there is unfortunately no log data to strengthen this result, it is nonetheless encouraging.

The logging data obtained suggests that the average time to build an entirely new *Assistment* is approximately 25 minutes. This data was acquired by examining the time that elapsed between the initialization of a new problem and the problem save time. Creation times for *Assistments* with more scaffolds naturally took longer than those with fewer scaffolds. Experience with the system also decreases *Assistment* creation time, as end-users who are more comfortable with the Assistment Builder are able work faster. Nonetheless, even users who were just learning the system were able to create *Assistments* in reasonable time. For instance, Users 2, 3, and 4 (see Table 1) provide examples of end-users who have little experience using the Assistment Builder. In fact, some of them are using the system for the first time in the examples provided.

Username	**Number of Scaffolds**	**Time Elapsed (min)**
User 1	10	35
User 1	2	23
User 2	3	45
User 2	2	31
User 2	0	8
User 3	2	21
User 4	3	37
User 4	0	15
User 5	4	30
User 5	2	8
User 5	4	13
User 5	4	35
User 5	3	31
User 5	2	24
		Average: 25.4 minutes

Table 1 - Full Item Creation

We were also able to collect useful data on morph creation time and Assistment editing time. On average morphing an *Assistment* takes approximately 10-20 minutes depending on the number of scaffolds in an Assistment and the nature of the morph.

4.0 Conclusions

The Assistment Builder has been in use over six months by a variety of users involved in the Assistments project. Teachers, developers, and others have used it to develop pseudo-tutor *Assistments*. The end result has been over a thousand individual pseudo-tutors deployed on the web. The breadth of users who developed these *Assistments* and the number created would not have been possible without the Assistment Builder.

References

1. Anderson, J. R. (1993). Rules of the mind. Hillsdale, NJ: Erlbaum.
2. Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences*, 4 (2), 167-207.
3. Jackson, G.T., Person, N.K., and Graesser, A.C. (2004) Adaptive Tutorial Dialogue in AutoTutor. *Proceedings of the workshop on Dialog-based Intelligent Tutoring Systems at the 7th International conference on Intelligent Tutoring Systems. Universidade Federal de Alagoas, Brazil, 9-13.*
4. Jarvis, M., Nuzzo-Jones, G. & Heffernan. N. T. (2004) Applying Machine Learning Techniques to Rule Generation in Intelligent Tutoring Systems. Proceedings of 7th Annual Intelligent Tutoring Systems Conference, Maceio, Brazil. Pages 541-553
5. Koedinger, K. R., Aleven, V., Heffernan. T., McLaren, B. & Hockenberry, M. (2004) Opening the Door to Non-Programmers: Authoring Intelligent Tutor Behavior by Demonstration. *Proceedings of 7th Annual Intelligent Tutoring Systems Conference, Maceio, Brazil.* Page 162-173
6. Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, *8,* 30-43.
7. Murray, T. (1999). Authoring intelligent tutoring systems: An analysis of the state of the art. International Journal of Artificial Intelligence in Education, 10, pp. 98-129.
8. Razzaq, L., Feng, M., Nuzzo-Jones, G., Heffernan, N.T., Aniszczyk, C., Choksey, S., Livak, T., Mercado, E., Turner, T.E., Upalekar. R, Walonoski, J.A., Macasek. M.A., Rasmussen, K.P. (2005) The Assistment Project: Blending Assessment and Assisting. *Submitted to the 12th Annual Conference on Artificial Intelligence in Education 2005, Amsterdam.*

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

What Did You Do At School Today? Using Tablet Technology to Link Parents to their Children and Teachers

Joshua UNDERWOOD, Rosemary LUCKIN, Lucinda KERAWALLA,
Benedict DU BOULAY,Joe HOLMBERG, Hilary TUNLEY, Jeanette O'CONNOR
IDEAs Laboratory, Department of Informatics, University of Sussex,
Brighton, BN1 9QH, UK

Abstract. The Homework project is developing an exemplar system for the delivery of adaptive, interactive numeracy and literacy education for children (5 to 7 year olds) at home and in the classroom. We illustrate how a user centred approach has enabled us to specify and go some way towards developing technology to meet the clear need for a better way of helping parents to engage with their children's homework and understand what they have done at school.

1 Introduction

The Homework project at Sussex University is developing an exemplar system for the delivery of adaptive, interactive numeracy and literacy education for children aged 5 to 7 years at home and in the classroom. It uses a combination of interactive whiteboard, tablet PC and wireless technology. The interactive whiteboard is for use in school; the tablet PCs can be taken home by each child for use with help from parents. Children watch TV quality video material, undertake exercises and project work as well as play games based on Number Crew materials for basic arithmetic [1]. The system is based upon the Broadband Learner Modeling pedagogical framework [2] and is the subject of a Human Centred Design methodology. We are combining theories and techniques from Narrative and from Artificial Intelligence and Education to produce a system that can scaffold a learner's progress in an adaptable and adaptive manner across multiple contexts and technologies [3-8].

2. Prototyping the Home-School link

In December 2004 we conducted an empirical evaluation of a limited prototype version of the Homework system (not linked to the learner modeling component) with a class of 32 six-year-old children and their teacher at School A. The Homework home interface used in the trial was developed in Flash, and links to Director, MPEG, Flash and PDF content, see Figure 1. From the home page it provides three views, a *history* of recent work through which relevant media can be revisited, *this week at school* with links to content used at school and *this week at home* with activities for home. The system logs all tablet PC activity for later analysis (and in due course for updating the learner model).

The study used an interactive whiteboard, wireless network and 5 tablet PCs in the classroom. On each weekday of the study we worked with a group of 5 children (selected by the teacher) who used the tablets for 20-30 minutes in class and took them home at the end of the day. The whiteboard was used with this group for video, singing and a polling application. The class teacher used the whiteboard for whole class video and interactive teaching programmes (ITPs). Each child was able take the tablet home for at least one night. Content was drawn from Number Crew material about division and multiplication. It fitted the curriculum objectives current for this class at the time of the study. The video and ITPs used by the teacher with the whole class were also available on the tablets.

Our study data include: classroom observers' notes including comments from the teacher and teaching assistants, parents' comments in diaries, usage logs from the Homework system and data entered (e.g. answers to exercises). Additional logging software (Activity Logger) captured information about how the tablets were used. We are currently analysing the data, but it is clear that the children were highly engaged by the content and were excited by and enjoyed the physicality of having their own tablets. The class teacher was also positive; he liked the integration of the technology and activities with his current numeracy work. Of the 29 diaries containing helper comments 24 contained sentences using 'fun/like/love/enjoy' using the system. Often this was explicitly linked to the idea that learning could be fun. 'Excited/couldn't wait/eager to use it' appears 14 times often coupled to the idea that the children were eager to do 'homework'. It was also clear that activity had often been in collaboration with a parent or other carer. Several parents mentioned helping the child with an activity and two diaries specifically mentioned a perceived improvement in the child's attention while doing Math activities on the tablet as opposed to on paper. Two diaries explicitly mention pleasure at being able to see what the child had been up to at school.

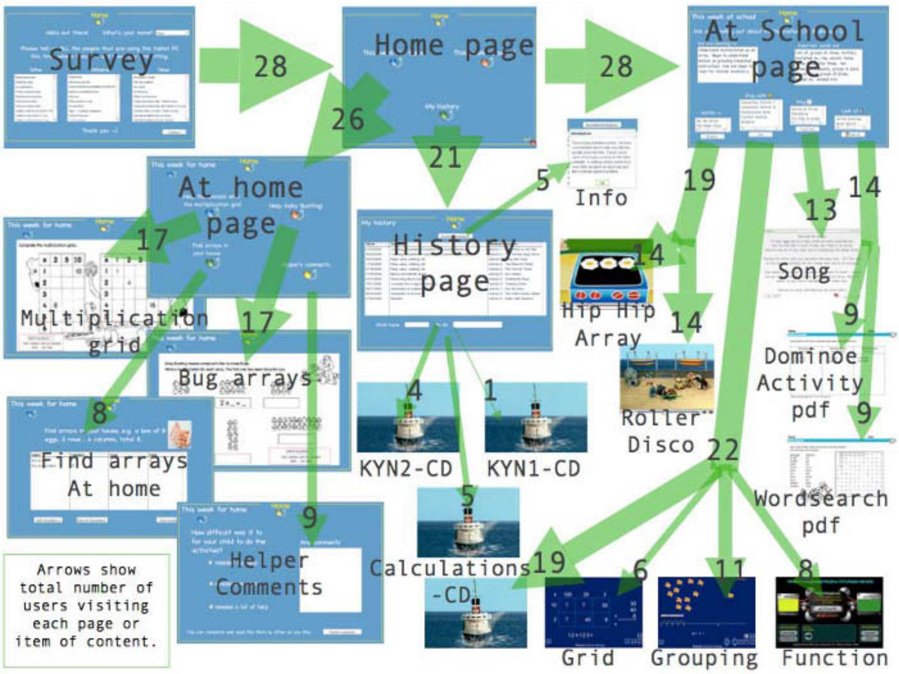


Figure 1. Homework 'home' interface – showing usage of the interface and content.

The following negative comments identified (the number in brackets indicates the number of occurrences of that comment): slowness (at start-up and clicking and waiting) (3), frustration with failed handwriting recognition (3 of these mention left handed issue) (5), specific difficulties with home or content interfaces (3), activities too easy (1), activities too difficult (1), made arm ache (1), it needs a stand (1), child doesn't want help (1)!

The log data from 28 sessions (4 logs were lost) illustrate that 100 percent of sample visited 'this week at school' and launched 1 or more pieces of linked content (see Figure 1). 93 percent visited 'this week at home' with 100 percent of these launching 1 or more pieces of content. 75 percent visited 'My history', though few of these actually launched activities.

3 Discussion and Conclusions

Our analysis has indicated that the large majority of parents and carers want to help their children with homework, that they are willing to devote time to this activity. All parents who were asked said that they would like to know more about what their children did at school each day: children don't want to or can't remember sufficient to tell their parents. Similarly, teachers expressed the desire for parents and carers to participate in learning activities at home and to follow their child's school learning experiences. Nevertheless, the practice and effectiveness of the links that already exist between home and school varies greatly. This should be a situation where technology can help, provided that it is designed to meet the users' needs and integrate with the users' context. Current school practices are components in the learning culture to which each learner belongs and must be taken into account when technology is introduced. The user centred design approach adopted throughout our work enables us to explore and map the context and participants for which the technology is being developed, it informs each phase of our system design: the development of the prototype technology and the empirical design used for the school study were informed by our previous interactions with teachers, parents and children.

Parents and teachers initial enthusiasm about the homework system illustrated in our early studies was confirmed when they were offered the opportunity to use the prototype system. All of the work described here has contributed to the detailed agenda of issues that we will fold into the development of the next iteration of the homework system during which we intend to develop further those parts of the system that provide assistance to the teacher in designing activities for the day, as well as the interface for home use that provides the all-important links between parents and what their children are doing at school.

Acknowledgements

We thank the teachers, parents and children for their help with this study which is funded by a EPSRC/ESRC/DTI PACCIT grant

References

[1.] Luckin, R., et al. *Coherence Compilation: Applying AIED Techiniques to the Reuse of Educational TV Resources*. in *Intelligent Tutoring Systems*. 2004. Maceio, Brazil. p. 98-107.

[2.] Luckin, R. and B. du Boulay, *Embedding AIED in ie-TV through Broadband User Modelling (BbUM)*, in *Proceedngs of 10th International Conference on Artificial Intelligence in Education*, J. Moore, W.L. Johnson, and C.L. Redfield, Editors. 2001, IOS Press: Amsterdam. p. 322-333.

[3.] Vygotsky, L.S., *Mind in society: the development of higher psychological processes*. 1978, Cambridge, MA: Harvard University press.

[4.] Wood, D. and H. Wood, *Vygotsky, tutoring and learning.* Oxford review of Education, 1996. **22**(1): p. 5-16.

[5.] Luckin, R. and B. du Boulay, *Ecolab: the Development and Evaluation of a Vygotskian Design Framework.* International Journal of Artificial Intelligence and Education, 1999. **10(2)**: p. 198-220.

[6.] Wood, D.J., J.S. Bruner, and G. Ross, *The role of tutoring in problem solving.* Journal of Child Psychology and Psychiatry, 1976. **17**(2): p. 89-100.

[7.] Jackson, S., J. Krajcik, and E. Soloway. *The Design of Guided Learner-Adaptable Scaffolding in Interactive Learning Environments*. in *Conference on Human Factors in Computing Systems*. 1998. Los Angeles, California, United States: ACM Press/Addison-Wesley Publishing Co. New York, NY, USA.

[8.] Tunley, H., du Boulay, B., Luckin, R., Holmberg, J., Underwood, J. Extending SCORM to model collaboration in contrasting school and home environments, Proceedings of User Modelling 2005, in press.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Semantic Description of Collaboration Scripts for Service Oriented CSCL Systems

Guillermo Vega-Gorgojo [1], Miguel L. Bote-Lorenzo, Eduardo Gómez-Sánchez, Yannis A. Dimitriadis and Juan I. Asensio-Pérez
School of Telecommunications Engineering, University of Valladolid, Spain

Abstract. Many CSCL systems have embraced scripting and service oriented computing to achieve effective collaboration and system flexibility, respectively. While learning standards, such as IMS-LD, can be used for scripting, we have encountered some problems to describe activity types, their collaboration properties and learning tools with this standard. The usability of collaboration scripts is limited, since some important features cannot be described. Furthermore, poor description of tools hinders the search of tools, offered as services, in service oriented CSCL systems. To overcome these difficulties, we propose an ontology that can be used to enrich the description of activities and tools in a script. Besides, the authoring process of a learning design is eased due to enforced restrictions in the ontology as well as the use of off-the-self ontology editors. Furthermore, formal and explicit semantics in a script can be exploited to automate the search of tools. This way, service providers can describe their tools in terms of the ontology, while educators can search for them using domain concepts.

1. Introduction

CSCL systems can be benefited both from *scripting* and *service oriented computing* [6]. "Scripting is a means to enhance the effectiveness of collaboration by prescribing how students should form groups, how they should interact and collaborate and how they should solve the problem" [3]. Scripts can be interpreted by systems in order to manage the sequence of activities to be performed by learners. Services can be employed in order to provide the software tools required to support a learning experience. An example of a CSCL system that adopts scripting and the service oriented approach is Gridcole [1]. It can be used as follows: learning designers can build their own scripts to model their educational scenarios. Next, a script interpreter will validate the script and arrange the sequence of activities. Then, external resources and tools offered as services needed to support the scenario described in the script will be discovered and integrated. Finally, users will join the resulting set up.

Developing such a system involves many challenging issues. First of all, an *Educational Modelling Language* (EML) is needed to formalize the collaboration scripts, so that it can be unambiguously interpreted. This way, a script player could manage the flow of activities to be performed in an educational system, as well as the arrangement of needed learning resources. The IMS Learning Design (IMS-LD) specification is, perhaps, the most relevant and complete EML for e-learning. Interestingly, it can be used to describe collaboration scripts although with some restrictions [5].

A collaboration script comprises a flow of activities that can be performed individually or collaborativelly. Each activity is supported by a set of learning resources of two types: tools and contents. Although the IMS-LD model uses these abstractions, we have encountered some difficulties when using IMS-LD to formalize collaboration scripts. First, *activity types are not defined*. Each activity type, e.g. an edition or a debate, has some distinguishing properties, such as specific outcomes and roles, that should be identified in a collaboration script. Since authoring a learning design is an error-prone and time-consuming task, an authoring system could embed this information to support the user when authoring a design. Second, *collaborative activities cannot be properly described* [5] because IMS-LD provides no means to specify how individuals collaborate within each activity. This is critical to state how learners should interact to perform a collaborative activity. A third issue is *the description of learning tools in a script*. IMS-LD can integrate descriptions of learning objects in a learning design using standards such as IEEE LOM or the service elements included in the IMS-LD specification (e.g. a conference). However, only a limited set of tools can be specified, as standards of learning objects do not even define a vocabulary of learning tools. On the one hand, these problems reduce the expressiveness of col-

[1] Correspondence to: Guillermo Vega-Gorgojo, School of Telecommunications Engineering, University of Valladolid, Camino del Cementerio s/n, 47011 Valladolid, Spain. Tel.: +34 983423000; Fax: +34 983423667; E-mail: guiveg@tel.uva.es

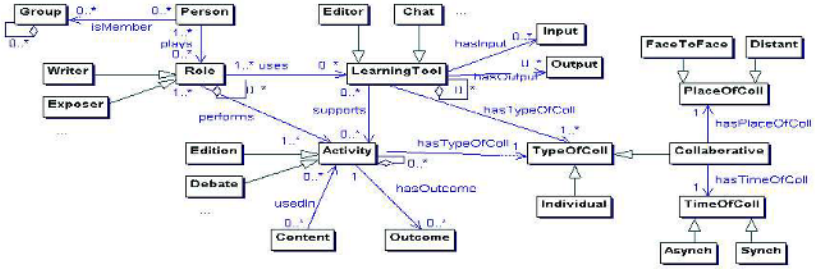


Figure 1. Conceptual model of an ontology of activities and learning tools. Core concepts are roles, activities and learning resources. The description of activities is decoupled from the description of tools and contents to achieve enhanced flexibility in the design.

laboration scripts precluding usability in educational scenarios and script sharing, since significant information cannot be expressed. On the other hand, poor description of learning tools severely limits automated discovery of appropriate tools, offered as services, to be integrated in a service oriented CSCL system. If learning tools were properly described in a learning design, a computer agent could support the discovery of suitable services. Otherwise, an educator should manually search for learning tools, reducing the usability of such CSCL system.

To overcome these difficulties, a collaboration script should better describe learning activities with meaningful information about activity types and collaboration features. Furthermore, describing the required learning tools to support a collaboration script would ease the binding of specific tools during the enactment of the scenario. In his sense, an ontology could be employed to capture the semantics involved in the description of learning activities and tools. Ontologies [2] are used to explicitly formalize knowledge in a shared manner, enabling rich descriptions and robust information retrieval systems. Thus, in this paper we propose an ontology that can be used to enrich the description of the activities and tools involved in collaboration scripts, while easing the authoring process. Besides, in a service oriented learning system tools offered as services can be searched using the learning abstractions described in the ontology. In previous work we analysed current service discovery mechanisms and proposed the use of educational ontologies to ease educators to search for learning services using their own terms [7].

In section 2 we describe an ontology that can be used to enrich the description of activities and learning tools involved in collaboration scripts. Section 3 illustrates the application of such ontology in a collaborative learning scenario. Finally, the main conclusions are shown.

2. Describing Collaboration Scripts with an Ontology of Activities and Learning Tools

IMS-LD has some important limitations to describe activities, specially collaborative activities. Besides, it is difficult to specify the tools required to support an activity. These facts limit the expressiveness of learning designs as well as the search of appropriate tools by educators. Both issues can be tackled by the semantic annotation of the activities and tools included in an IMS-LD-compliant script. An ontology can be employed to formalize this required knowledge with explicit semantics which can be easily shared and it is interpretable by the learning infrastructure.

A feasible model of such an ontology is shown pictorially in figure 1. The problem of specifying activity types is tackled defining a set that can be applied to a broad range of collaboration scenarios, such as *Debate* an *Edition*. Second, collaboration capabilities defined at activity and tool levels can be expressed using this ontology. The well-known categorization using time and space [4] is employed here. Finally, learning tools such as *Editor* or *Chat* can be described using the educational abstractions modelled in the ontology.

3. Application in a Collaboration Learning Scenario

To illustrate the application of the proposed ontology, a simple collaborative scenario based on the well-known “snowball” collaboration pattern is described using the ontology abstractions, shown in table 1. Although this script can be formalized in IMS-LD, problems detected in section 1 should be addressed in order to enable the actual realization of the scenario. This way, a semantic description of the involved tools and activities is provided and can be attached to the IMS-LD script to enable the unambiguous interpretation of the script.

While service oriented computing advocates increased flexibility and reusability to deliver software, it introduces the problem of discovering appropriate services in order to realize such systems. In the case of service

Table 1. Description of a sample collaborative learning scenario. It comprises three sequential activities: A1 consists on reading a document, in A2 learners must individually respond to a questionnaire about the document, while A3 depicts a collaborative debate in which participants have to agree to a common response. These activities, as well as the contents and tools that support them, are described using the abstractions modelled in the proposed ontology, shown in figure 1.

Activity					Content		Tool					
Ref	**Type**	**Collab**	**Roles**	**Outc**	**Ref**	**Roles**	**Ref**	**Type**	**Collab**	**Roles**	**In**	**Out**
A1	Studying	Indiv	Learner	-	D1	Learner	T1	Viewer	Indiv	Learner	D1	-
A2	Assessment	Indiv	Submitter	D3	D2	Submitter	T2	Questionnaire	Indiv	Submitter	D2	D3
A3	Debate	Collab Sync Distant	Debater Submitter	D4	D2	Submitter	T3	Chat	Collab Sync	Debater	-	-
					D3	Debater	T4	Viewer	Indiv	Debater	D3	-
							T5	Questionnaire	Indiv	Submitter	D2	D4

oriented CSCL systems, educators are usually in charge on setting the arrangement of the scenario, including the search of tools. They should be capable to perform this search in a convenient way. Therefore, educators could use the educational abstractions formalized in the proposed ontology to search for tools if providers commit to this ontology. An extensive discussion about this topic is offered in [7].

In the depicted scenario, a computer agent can interpret the semantic tool descriptions in the script (tools *T1* through *T5* in table 1) and query registries of learning tools for the providers descriptions. Educators can use the encountered tools or begin a new query using the ontology concepts.

4. Conclusions and Future Work

Current educational standards for scripting have some limitations to describe collaboration scripts. The ontology proposed in this paper overcomes these problems enabling the semantic description of these features, while still conforming to existing standards, such as IMS-LD. This way, educational scenarios can be deeply described allowing for enhanced usability, since the underlying learning infrastructure can take appropriate actions to enact the scenario. Besides, semantic description of tools and activities can be exploited to automate the search of tools, offered as services, in a service oriented CSCL system.

Acknowledgements

This work has been partially funded by Spanish Ministry of Science and Technology project TIC-2002-04258-C03-02, the European Commission e-learning project EAC/61/03/GR009 and the Kaleidoscope network of excellence FP6-2002-IST-507838. Special thanks to the MOSIL research project for the MOSIL deliverable 2.

References

[1] M.L. Bote-Lorenzo, L. Vaquero-González, G. Vega-Gorgojo, Y. Dimitriadis, J. Asensio-Pérez, E. Gómez-Sánchez, and D. Hernández-Leo. A tailorable collaborative learning system that combines OGSA grid services and IMS-LD scripting. In *Proceedings of the Tenth International Workshop on Groupware: Design, Implementation, and Use (CRIWG 2004)*, LNCS 3198, pages 305–321, San Carlos, Costa Rica, Sept. 2004. Springer-Verlag.

[2] B. Chandrasekaran, J. Josephson, and V. Benjamins. What are ontologies, and why do we need them? *IEEE Intelligent Systems*, 14(1):72–81, 1999.

[3] P. Dillenbourg. Over-scripting CSCL: the risks of blending collaborative learning with instructional design. In P. Kirschner, editor, *Three worlds of CSCL. Can we support CSCL?*, pages 61–91. Open Universiteit Nederland, Heerlen, 2002.

[4] C. A. Ellis, S. J. Gibbs, and G. L. Rein. Groupware: Some issues and experiences. *Communications of the ACM*, 34(1):38–58, 1991.

[5] D. Hernández-Leo, J. Asensio-Pérez, and Y. Dimitriadis. IMS learning design support for the formalization of collaborative learning patterns. In *Proceedings of the 4th International Conference on Advanced Learning Technologies (ICALT'04)*, pages 350–354, Joensuu, Finland, Aug. 2004.

[6] M. Papazoglou and D. Georgakopoulos. Service-oriented computing. *Communications of the ACM*, 46(10):25–28, Oct. 2003.

[7] G. Vega-Gorgojo, M.L. Bote-Lorenzo, E. Gómez-Sánchez, Y. Dimitriadis, and J. Asensio-Pérez. Semantic search of learning services in a grid-based collaborative system. In *Proceedings of the Fifth IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2005), Workshop on Collaborative Learning Applications of Grid Technology and Grid Education (CLAG + Grid.edu 2005)*, Cardiff, UK, May 2005. Accepted for publication.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

What's in a rectangle? An issue for AIED in the design of semiotic learning tools

Erica DE VRIES
Laboratory for Educational Sciences, Pierre-Mendès-France University & IUFM, Grenoble
BP 47, F-38040 Grenoble cedex 9, France. Erica.deVries@upmf-grenoble.fr

Abstract. This poster presents a tentative analysis of the conventions for a rectangle as a widespread graphical element in computer-based learning tools.

1. Introduction

Nowadays, computers are thought of as cognitive tools for learners to construct knowledge by actively building and manipulating, interpreting and negotiating. Three working assumptions found their design [1, 2]: learning involves *construction* of one's own representations rather than merely interpreting prefabricated representations, *translation* and *manipulation* of multiple representations, and *interaction* with others to negotiate meaning. In a nutshell, the computer is put forward as a semiotic tool with representational affordances for construction, manipulation and collaboration as key learning activities. In this poster, I take the notion of semiotic learning tools to the letter. Semiosis refers to the process of making meaning out of objects, symbols, or anything else perceived in the environment. Semioticians conceive of humans as evolving in many interrelated sign systems, i.e. in a large complex system of sign systems [3]. Semiotics, as the study of signs, investigates the cultural conventions that shape the relation between signifiers (representing) and what is signified (represented). Learning tools, when qualified as semiotic, supposedly help learners in meaning making by depending on existing, evolving or emerging conventions regarding graphical and linguistic elements on computer screens. The poster presents a tentative analysis for a particularly pervasive graphical element: a rectangle.

2. About representation

A representation is often defined as *something that stands for something else*, i.e. internal representations in the human mind stand for objects and relations in the real world [4]. Learning can be conceptualised as the construction of internal representations on the basis of external ones and recent research focuses more explicitly on the interplay of internal and external representations [5]. Semiotics, in addition, takes into account both the nature of the situation and of the entity that is interpreting; cf. one of Peirce's definitions of a sign: *something which stands to somebody for something in some respect or capacity* [6]. In this view, the meaning of a representation depends on cultural conventions and on person, task and situation characteristics. The case for semiotics is threefold: meaning making is an important educational goal, domain conventions are often part of learning objectives, and learning entails becoming independent of a particular representation which entails manipulation of several different representations. Relevant dimensions in education therefore appear to be the degree of domain-specificity and familiarity of representations and the amount of required translation between representations.

3. Rectangles in computer-based learning tools

Rectangles are everywhere on computer screens and experienced users easily interpret a rectangle as a label, a cell, a widget, a button, or anything else, except maybe in using an unfamiliar program. According to general conventions, a rectangle signifies a *label* when it encloses other symbols (a road sign with the name of a city), a *container* when it is supposed to hold something (a 2D projection of a box), a *building block* when it is a 2D projection of a solid, a *location* when spatial configuration matters, or a *process* when it incarnates some transformation with an input and output. The aim of the analysis presented here is to identify emerging conventions in computer tools for learning.

3.1 Concept maps and hypermedia construction tools

Concept maps, like semantic networks, depict knowledge in terms of concepts (the nodes) and relations (labelled lines or arrows). In order to construct one, learners need to identify and explicit concepts and their interrelationships. The same kind of maps, either learner or teacher constructed, are also used as organizers of hypermedia material, cf. a web-view as a dynamic clickable map. In such semantic organization tools, rectangles signify *labels* of concepts or content. Although the tools generally use a code for distinguishing concepts from relations, the use of a different graphical element for labelling, e.g., ellipses instead of rectangles, would not change the intended interpretation or meaning of a map in most cases. Thus, the conventions regarding the use of rectangles are not domain specific and largely follow general cultural conventions.

3.2 Modelling and simulation tools

A second category concerns modelling and simulation tools for learning in biology, physics, etc. E.g., *Stella* uses a highly specific code: a rectangle signifies a *stock* of something, clouds are resources, and circles are flows or constants. In most modelling tools however, rectangles signify *variables* (and relations) much like in concept maps. The difference is that modelling tools allow entering a system of equations to represent the underlying mathematical model, an activity that requires translation between representations. *Boxer* uses rectangles to signify *procedures* (e.g. “move s”) of a certain type (e.g. “Doit”) and that have a rectangle attached as a *label* with its name (e.g. “tick”). Boxer also has rounded rectangles for *data*, but no lines or arrows to represent relations or flows. These examples show highly specialized yet different local and relatively unfamiliar conventions. Despite the fact that they concern mathematical models of dynamic systems, these tools do not seem to conform to a common established domain-specific convention.

3.3 Collaboration and discussion tools

In *Belvedere*, rectangles are *data* (facts, observations) and rounded rectangles are hypotheses, arrows are relations with a colour code, red for supporting (in favour of) and green for invalidating (against) relations. In *Drew*, rectangles are *non-conflictual propositions*, squeezed rectangles are conflictual ones, circles are non-conflictual relations and diamonds are conflictual ones, a colour code signals contributions of different participants. Whereas the *Belvedere* format stresses epistemological stance (hypothesis, data), the *Drew* format stresses individual contributions and (dis)agreement. Moreover, squeezed rectangles in *Drew* seem to hinge on the connotation of being wedged or wicked to signify disagreement. Both examples show local conventions independent of a particular domain and relatively unfamiliar to the learners.

4. Conclusions

External representations that conform to a formal code in the eyes of the constructor, may not necessarily be univocal to the individual that reads and manipulates them in a particular situation. The tentative analysis on rectangles shows that we cannot yet conceptualize semiotic learning tools as being part of a unique semiological system [7]. Although the mode of operation (visual, graphical) and the domain of validity (learning situations) are similar, both the nature and number of signs (rectangles, arrows, circles, texts) and the type of functioning (presence/absence, simultaneity, location) vary considerably from tool to tool.

The presented tools use conventions for rectangles that are little domain-specific, mostly unfamiliar to the learner, and, with the exception of modelling and simulation tools, do not require much translation between representations. Unspecific conventions, i.e., that are not grounded in a domain of expertise, may constitute an advantage, but an issue is whether learning will be robust when the learners' interpretation of graphical elements deviates from the intended meaning. Moreover, although they somewhat conform to general cultural conventions, the tools also introduce highly-specialized unfamiliar representations. In fact, they suggest a strong modelling perspective of knowledge construction as individual, mathematical or social activity. A second issue is therefore whether learners will easily adopt them. Finally, the tools require learners to adapt to a given representation, and do not particularly invite learners to translate between them (except for modelling tools). The question here is whether learners will be able to effortlessly switch from one representation to another in using more than one tool, and more importantly whether their knowledge construction will be independent of the particular representation used.

An implication of the analysis is we should be reluctant to qualify newly developed learning tools as semiotic. On the one hand, multiplying representational formats might be a source of confusion given that users are learners of representational formats as much as they are learners of content. A set of local, unfamiliar, unsanctioned and incoherent representational formats in this view would be semiotic obstacles rather than semiotic tools. On the other hand, representational diversity could also remain unnoticed precisely because humans are thought to evolve in a complex system of multiple sign systems anyway. In this perspective, humans are trained interpreters of and adapters to all sorts of external representations even in a learning situation. In the latter case, speaking of semiotic learning tools implicitly carries a denial of the pertinence of particular representational formats.

References

[1] Ainsworth, S. (1999). The functions of multiple representations. *Computers and Education, 33*, 131-152.

[2] Duval, R. (1995). *Sémiosis et pensée humaine* [Semiosis and humain thought]. Bern: Peter Lang.

[3] Eco, U. (1988). *Le signe, histoire et analyse d'un concept* [The sign, history and analysis of a concept]. Translated from Italian by J.-M. Klinkenberg. Bruxelles: Editions Labor.

[4] Palmer, S. E. (1978). Fundamental aspects of cognitive representation. In E. Rosch & B. B. Lloyd (Eds.), *Cognition and categorization* (pp. 259-203). Hillsdale, NJ: Lawrence Erlbaum Associates.

[5] Zhang, J. & Norman, D. A. (1994). Representations in distributed cognitive tasks. *Cognitive Science, 18*, 87-122.

[6] Peirce, C. S. (1978). *Collected Papers 1331-1335*. Cambridge: Harvard University Press. (Partially translated by G. Deledalle (Ed.), Charles S. Peirce : Ecrits sur le signe. Paris : Editions du Seuil).

[7] Barthes, R. (1964). Eléments de sémiologie [Elements of semiology]. *Communications, 4*, 91-135.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

A User Modeling Framework for Exploring Creative Problem-Solving Ability

Hao-Chuan WANG[1], Tsai-Yen LI[2], and Chun-Yen CHANG[3]

Institute of Information Science, Academia Sinica, Taiwan[1]
Department of Computer Science, National Chengchi University, Taiwan[2]
Department of Earth Sciences, National Taiwan Normal University, Taiwan[3]
haochuan@iis.sinica.edu.tw, li@nccu.edu.tw, changcy@cc.ntnu.edu.tw

Abstract. This research proposes a user modeling framework which aims to assess and model users' creative problem-solving ability from their self-explained ideas for a specific scenario of problem-solving. The proposed framework, User Problem-Solving Ability Modeler (UPSAM), is mainly designed to accommodate to the needs of studying students' Creative Problem-Solving (CPS) abilities in the area of science education. The use of open-ended essay-question-type instrument and bipartite graph-based modeling technique together provides a potential solution of user model elicitation for CPS. The computational model has several potential applications in educational research and practice, including automated scoring, buggy concepts diagnosis, novel ideas detection, and supporting advanced studies of human creativity.

1. Introduction

Problem-solving has consistently been an attractive topic in psychological and educational research for years. It is still a vital research field nowadays, and its role is believed to be much more important than it used to be, in alignment with the trends of putting stronger emphasis on students' problem-solving process in educational practices.

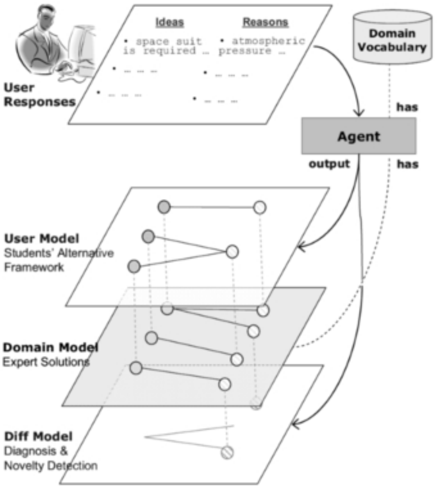
User Modeling (UM) for problem-solving ability is an alluring and long-going research topic. Previous works in the area of Intelligent Tutoring Systems (ITS) have endeavoured substantially to model problem-solving process for well defined problem contexts, such as planning a solution path in proving mathematical theorems or practicing Newtonian physics exercises [3]. However, we think the classical ITS paradigm *cannot* well describe the process of *divergent* and *convergent thinking* in the human Creative Problem-Solving (CPS) tasks [1][5]. In other words, the classical approach lacks the functionality to support advanced educational research on the topic of CPS.

In this paper, we propose a user modeling framework, named UPSAM (User Problem Solving Ability Modeler), by exploiting *open-ended essay-question-type instrument* and *bipartite graph-based representation* to capture and model the creative perspective of human problem-solving. UPSAM is designed to be flexible and can have several potential advantageous applications, including: 1) offering functionalities to support educational studies on human creativity, such as automated scoring of open-ended instruments for CPS, and 2) detecting students' alternative conception on a particular problem-solving task for enabling meta-cognitive concerns in building adaptive educational systems.

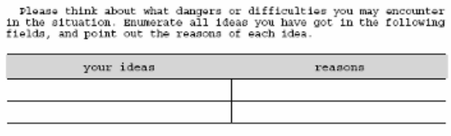
2. UPSAM: User Problem Solving Ability Modeler

A bird's eye view of the UPSAM framework is abstractly depicted in Figure 1. The grey box labelled *Agent* refers to the core software module implemented several functionalities to perform each process of user modeling as described in [4], including:

1) Perceiving the raw data from the user (the process of eliciting user information),
2) Summarizing the raw data as the structured user model (the process of monitoring/modeling), and
3) Making decisions based on the summarized user model (the process of reasoning).



Note that the source data for UPSAM are users' free-text responses in natural language toward an open-ended essay-question-type instrument. However, although users' responses are open-ended, they are *not* of no structure by themselves. With the help of a controlled *domain vocabulary* which increases the consistency between users' and the expert's wording, as well as the *pair-wise semi-structured nature* of the instrument which help identify the context of users' answers, it becomes much more tractable to perform the operation of user model summarization from such open-ended answers.



Please think about what dangers or difficulties you may encounter in the situation. Enumerate all ideas you have got in the following fields, and point out the reasons of each idea.

your ideas	reasons

Figure 2. A snapshot of the answer sheet showing the pair-wise relation among ideas and reasons.

Figure 2 depicts the format of the instrument for eliciting user information, which is based on the structure of the CPS test proposed by Wu *et al.* in [5]. Users are required to express their ideas (cf. the production of divergent thinking in CPS) in the problem-solving context described by the instrument, and then explain/validate each idea with reasons (cf. convergent thinking in CPS).

3. Bipartite Graph-based Model

In UPSAM, an important feature to capture users' CPS ability is to structure the domain and user models (see Figure 1) as bipartite graphs. Actually, a domain model is simply a special case of user model summarized from domain experts with a different building process. Domain models are now authored by human experts manually, while user models are built by UPSAM automatically. Therefore, the fundamental formalism of the domain and user models is identical.

One of the most important features in CPS is the relation bewteen divergent thinking and convergent thinking. The bipartite graph in the graph theory is considered appropraite to represent this feature. A bipartite graph is one whose vertex set can be partitioned into two disjoint subsets such that the two ends of each edge are from different subsets [2]. In this case, given a set of ideas $A=\{a_1, a_2, \ldots, a_n\}$ and a set of reasons $B=\{b_1, b_2, \ldots, b_m\}$, the domain model can be represented as an undirected bipartite graph $G=(V, E)$ where $V=A \cup B$ and

$A \cap B = \phi$. The connections between ideas and reasons are represented as $E=\{e_{ij}\}$, and each single edge e_{ij} represents a link between idea a_i and reason b_j.

Different ideas, reasons, and combinations of the *(idea, reason)* pairs should be given different scores indicating the quality of answers. The scoring functions are assigned to *A*, *B*, and *E*, respectively:

$$Sc = \{good\ answer,\ regular,\ no\ credit\},\ f_A : A \to Sc, f_B : B \to Sc, \text{and } f_E : E \to Sc$$

where S_C denotes the range of these scoring functions, and each ordinal value (ex. "*regular*") is connected to a corresponding numeric value. Then the total score of a model $G=(A \cup B, E)$ can be computed as the weighted summarization of individual part of scores:

$$f_{total}(G) = (w_A f_A(A) + w_B f_B(B) + w_E f_E(E))/(w_A + w_B + w_E)$$

w_A, w_B, and w_E are weighting coefficients that can be tuned according to the needs of each application. Therefore, the score for a user *U* can be reasonably defined as the ratio of the user model's (G_U) total score to the domain model's (G_D) total score. That is, $Score(U)=f_{total}(G_U)/f_{total}(G_D)$. An automated scorer for grading semi-structured responses can then be realized accordingly. Moreover, a fine grained analysis of users' cognitive status is possible by considering the difference between the domain and user models. The *Diff Model* representing the difference is defined as $G_{diff}=(G_U \cup G_D) - (G_U \cap G_D)$. Its properties and applications deserve further exploration.

The process of building the bipartite graph-based user models from users' answers is computationally tenable. The kernel idea is to employ techniques of Information Retrieval (IR) to identify the similarity between users' open-ended entries and the descriptions associated to each vertex in the domain model. As mentioned in Section 2, the incorporation of a controlled vocabulary and the structure of the instrument are considered helpful to the process. A prototypical automated user modeling and scoring system has been implemented, and more details will be reported soon.

4. Conclusion

In this paper, we have briefly described a user modeling framework for CPS ability, UPSAM. Empirical evaluations, full-fledged details, and applications of the framework are our current and future works. We also expect that the computational model can be of contribution to the study of human creativity in the long run.

References

[1] Basadur, M. (1995) Optimal Ideation-Evaluation Ratios. *Creativity Research Journal*, Vol. 8, No. 1, pp.63-75.

[2] Boundy, J., Murty, U.S.R. (1976) *Graph theory with applications*, American Elsevier, New York.

[3] Conati, C., Gertner, A.S., VanLehn, K., and Druzdzel, M.J. (1997) On-Line Student-Modeling for Coached Problem Solving Using Bayesian Network. *Proceedings of 6th International Conference on User Modeling,* Italy.

[4] Kay, J. (2001) User Modeling for Adaptation. *User Interface for All: Concepts, Methods, and Tools*, Lawrence Erlbaum Associates, pp. 271-294.

[5] Wu, C-L., Chang, C-Y. (2002) Exploring the Interrelationship Between Tenth-Graders' Problem-Solving Abilities and Their Prior Knowledge and Reasoning Skills in Earth Science. *Chinese Journal of Science Education,* Vol. 10, No. 2, pp. 135-156.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Adult Learner Perceptions of Affective Agents: Experimental Data and Phenomenological Observations

Daniel WARREN
Instructional Systems Program
RITL – Affective Computing
Florida State University
rdw4048@fsu.edu

E SHEN
Instructional Systems Program
RITL – Affective Computing
Florida State University
ess0086@fsu.edu

Sanghoon PARK
Instructional Systems Program
RITL – Affective Computing
Florida State University
ssp5177@fsu.edu

Amy L. BAYLOR
Director, RITL
http://ritl.fsu.edu
Florida State University
baylor@coe.fsu.edu

Roberto PEREZ
Instructional Systems Program
RITL – Affective Computing
Florida State University
rgp6722@mailer.fsu.edu

Abstract. This paper describes a two-part study of animated affective agents that varied by affective state (positive or evasive) and motivational support (present or absent). In the first study, all four conditions significantly improved learning; however, only three conditions significantly improved math self-efficacy, the exception being the animated agent with evasive emotion and no motivational support. To help in interpreting these unexpected results, the second study used a phenomenological approach to gain an understanding of learner perceptions, emotions, interaction patterns, and expectations regarding the roles of agent affective state and motivational support during the learning process. From the qualitative data emerged three overall themes important to learners during the learning process: learner perceptions of the agent, learner perceptions of self, and learner-agent social interaction. This paper describes the results of the phenomenological study and discusses the findings with recommendations for future research.

1. Introduction

Animated agents are graphical interfaces that are capable of using verbal and non-verbal modes of communication to interact with users in computer-based environment. These agents generally present themselves to users as believable characters, who implement a primitive or aggregate cognitive function by acting as mediators among people and programs, or by performing the role of an intelligent assistant [1]. In other words, they simulate a human relationship by doing something that another person could otherwise do for that user [2]. There has been extensive research that shows learners in agent-based environments have showed deeper learning and higher motivation [3]. A recent study [4] in which agents monitored and evaluated the timing and implementation of teaching interventions, has indicated that agent role and agent voice and animation had a positive effect on learning, motivation, and self-efficacy. Yet, there are few studies which focus on the cognitive function of the agent in the learning environment [5], or which implement a systematic examination of learner motivation, perceived agent values, and self-efficacy. The focus of this study is to explore how users perceive emotionally evasive and unmotivated agents, and to try to uncover what perceptions and alternative strategies users may develop to deal with this kind of agent.

2. Experimental Method

Sixty-seven General Education Development students in a community college in the southeastern United States participated in this study. Students were 52% male with 17.9% Caucasians, 71.6% African-Americans, and 13.5% of other ethnicities, with average age 22.3 years (SD=8.75).

There were four agent conditions: 1) Positive affective state + motivational support; 2) Evasive affective state + motivational support; 3) Positive affective state only; 4) Evasive affective state only. Students were randomly assigned to one of the agent conditions, and they learned to solve percentage word problems. Before and after the task, students' math anxiety level and math self-efficacy were measured. The post-test also measured perceived agent value, instructional support, and learning.

3. Findings

Results indicated that students who worked with the positive + motivation support agent significantly enhanced their self-efficacy from prior (M=2.43, SD = 1.22) to following the intervention (M = 3.79, SD = 1.37, p < .001). Similar improvement was found for the agent with positive affective state only (M=2.42, SD = .96 vs. M = 3.84, SD = 1.43, p < .001) and for the agent with evasive + motivation support (M = 3.06, SD = 1.53 vs. M = 4.13, SD = 1.03, p < .001). Additionally, students perceived the agent with motivational support as significantly more human-like (M = 3.83, SD = 1.02) and engaging (M = 4.03, SD = 1.09) than the agent without motivational support (M = 3.33, SD = 1.02) (M = 3.65, SD = .92). As expected, the agent with evasive affective state and no motivation support did not lead to an improvement of student self-efficacy or to a perception of the agent as offering good instructional support. However, across all conditions, students performed significantly better on the learning measure than prior to using the program. In other words, students who interacted with an emotionally evasive, un-motivational agent, still improved their learning (i.e., "in spite of" this agent). This result was intriguing enough to motivate the second part of the study, where students were observed and interviewed about their interactions with an agent that displayed evasive emotions and provided no motivational support. The focus of this part, then, was on understanding those interactions better, as well as getting students' feedback to improve the agent.

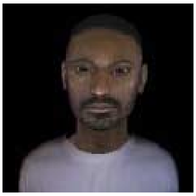


Fig. 1: the animated agent used in the stud

4. Observational Method

The phenomenological follow-up study included six students enrolled in an Adult Education program at the same southeastern United States community college. Participants were selected using intensity sampling to identify individuals willing to express opinions and describe their experiences.

Data were collected using direct observations and interviews. During the initial observation phase, participants navigated through a computer-based math learning module and interacted with a pedagogical agent that displayed evasive emotion without motivational support. Participants were asked at specific times to describe their perception of the agent's emotional expressions. Researchers observed participants from a control booth through one-way windows and took field notes noting participants' emotional expressions. During the follow-up interview, participants viewed digitally cued segments of their interactions with the agent, and were asked to describe their emotional expressions, feelings, and reactions at the specific time in the video recording.

4.3 Coding the Data

Coding the data involved looking for meaningful patterns and themes that aligned with the purposes and the focus of the study. Interview data were digitized and transcribed then imported into NVivo™ software for subsequent data coding and analysis.

4.4 Validation and Triangulation Process

Triangulation of findings involved: comparing field notes from observations, interviews, and survey responses; using different data collection methods; using different sources; and using perspectives from different analysts to review the data; which together lent further credibility to the findings.

5. Findings

From iterative and immersive data analyses emerged themes, each of which is discussed below.

Learner Perception of the Agent. This theme refers to learners' reaction toward the agent's: emotion, facial expression, gaze, image, voice, and initial reaction. Responses such as "it was strange," "what's going on," and "funny looking" characterize the initial reactions that students had toward the agent. Categories within this theme contained two sub-categories: "learner's assessment" (of the agent) and "learner's recommendation" (to improve the agent), both in regard to the agent's emotional expressions, facial expressions, and tone of voice.

Learner Perception of Self. This theme refers to learner: nervousness, anxiety, confusion, frustration, and confidence while interacting with the agent. Two categories not related to agent interactions but included in this theme were participants' emotional experience when exposed to timed questions, and learners' assessment of their prior content knowledge.

Learner-Agent Social Interaction. This theme refers to the agent's: feedback, overall nature and manner, and support and encouragement. Other emergent categories include: descriptions of possible agent social interaction interface options, favorite teacher characteristics, and descriptive comparisons of the agent versus a face-to-face teacher, and the agent's voice versus the screen text.

7. Conclusions

Participant responses imply that benefits of the agent depended on the learner and context characteristics. Participants seemed to perceive that having the agent present and interacting with them could have afforded the possibility for providing support for their learning, but that the specific instructional and support strategies with this particular agent did not always do so.

Participant suggestions in terms of agent voice quality, facial expressions, eye contact, gestures, and emotional responses can be used to improve the interface. These improvements also apply to learner's expectations for social interactions that do not distract from the learning task.

Participant responses also suggest that a more responsive agent in terms of the variety of learners' instructional needs would facilitate better learning experiences, and lead to less frustration and greater satisfaction. Participants expressed similar sentiments in terms of the agent's ability to provide more positive and reinforcing feedback and support, rather than simply saying "correct" or "incorrect," saying instead "good job" or "good try, but next time try better."

Although these results did not provide enough data to account for student gains in learning under unfavorable conditions (e.g., an agent with evasive emotional states), the study provided an insight into how students' emotions and perceptions developed in their interaction with an agent. At the same time, the experimental part of the study confirmed previous findings as to the benefits of motivational support and positive emotion displayed by an animated agent. Future research can be carried out on affect and how different aspects of the agent interact to affect the user.

8. Acknowledgements

This work was supported by the National Science Foundation, Grant IIS-0218692.

References

1.Bradshaw, J.M. Software agents. in Bradshaw, J.M. ed. An introduction to intelligent agents, MIT Press, Menlo Park, CA, 1997, 3-46.

2.Seiker, T. Coach: A teaching agent that learns. Communication of the ACM, 37 (7). 92-99.

3.Moreno, R., Mayer, R.E. and Lester, J.C., Life-Like Pedagogical Agents in Constructivist Multimedia Environments: Cognitive Consequences of their Interaction. in World Conference on Educational Multimedia, Hypermedia, and Telecommunication (ED-MEDIA), (Montreal, 2000).

4.Baylor, A.L. Permutations of control: cognitive considerations for agent-based learning environments. Journal of interactive learning research, 12 (4). 403-425.

5.Baylor, A.L. The effect of agent role on learning, motivation, and perceived agent value. Journal of Educational Computing Research.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Factors Influencing Effectiveness in Automated Essay Scoring with LSA

Fridolin Wild, Christina Stahl, Gerald Stermsek, Yoseba Penya, Gustaf Neumann
Department of Information Systems and New Media,
Vienna University of Economics and Business Administration (WU Wien),
Augasse 2-6, A-1090 Vienna, Austria
{firstname.lastname}@wu-wien.ac.at

Abstract. Automated essay scoring by means of latent semantic analysis (LSA) has recently been subject to increasing interest. Although previous authors have achieved grade ranges similar to those awarded by humans, it is still not clear which and how parameters improve or decrease the effectiveness of LSA. This paper presents an analysis of the effects of these parameters, such as text pre-processing, weighting, singular value dimensionality and type of similarity measure, and benchmarks this effectiveness by comparing machine-assigned with human-assigned scores in a real-world case. We show that each of the identified factors significantly influences the quality of automated essay scoring and that the factors are not independent of each other.

Introduction

Computer assisted assessment in education has a long tradition. While early experiments on grading free text responses had mostly been syntactical in nature, research today focuses on emulating a human-semantic understanding (cf. [12]). In this respect, Landauer et al. [1] found evidence that a method they named 'latent semantic analysis' (LSA) produces grade ranges similar to those awarded by human graders. Several stages in this process leading from raw input documents to the machine assigned scores allow for improvement. Contradicting claims, however, question the optimisation of these influencing factors (e.g. [2] vs. [9]).

In this contribution we describe an experiment on the optimization of influencing factors driving the automated scoring of free text answers with LSA. By testing automated essay scoring for the German language and through the use of a small text-corpus we extend previous work in this field (e.g. by [2, 3]). Whereas a detailed description of LSA in general can be found elsewhere (e.g. [1]), the following sections give an overview of the methodology, hypotheses and the results of our experiments.

1. Methodology

Formally, an experiment tries to explore the cause-and-effect relationship where causes can be manipulated to produce different effects [4]. In this way, we developed a software application to alter the settings of the influencing factors we adopted for an experimental approach. This enabled us to compare machine-assigned scores (our dependent variables) to the human-assessed scores by measuring their correlation, a testing procedure commonly used in the literature of essay scoring (e.g. in [5], [6], [7]). By changing consecutively and *ceteris paribus* the influencing factors (our independent variables), we investigated their influence on the score correlation.

The corpus of the experiment consisted of students' free-text answers to the same marketing exam question. The 43 responses were pre-graded by a human assessor (say, a teacher) with points from 0 to 5, assuming that every point was of the same value and thus, the scale was equidistant in its value representation. The average length of the essays was 56.4 words, a value that is on the bottom of recommended essay length [8]. From those essays that received the highest scores from the human evaluator, we chose three so-called ‘golden essays’. These golden essays were used to compute the correlation for the remaining essays assuming that a high correlation between a test essay and the mean of the golden essays entails a high score for the test essay [1]. The SVD co-occurrence matrix was built with the three golden essays and a marketing glossary consisting of 302 definitions from the domain of the exam. Every glossary entry was a single file with an average length of 56.1 words and the glossary was part of the preparation materials for the exam.

2. Hypothesis and Test Design

We conducted several tests addressing four aspects that have proven to show great influence on the functionality and effectiveness of LSA [1,2]:

1. *Document pre-processing:* With the elimination of stop-words and stemming in mind, we used a stop-word list with 373 German terms and Porter’s Snowball stemmer [11]. We assessed the effects of pre-processing by testing the corpus with and without stemming, with and without stop-word removal and with the combination of stemming and stop-word removal. For the succeeding tests, we used the raw matrix as default.

2. *Weighting-schemes:* Several weighting-schemes have been tested in the past (e.g. in [3, 9]), yielding best results for the logarithm (local weighting), and the entropy (global). Assuming that these results will also apply to the German language and the automated scoring of essays, we combined three local (raw term-frequency, logarithm, and binary) and four global (none, normalization, inverse document-frequency, and entropy) weightings. As default we used the raw term frequency and no global weighting.

3. *Choice of dimensionality:* The purpose of reducing the original term-document matrix is to minimize noise and variability in word usage [10]. In order to determine the amount of factors needed for the reduced matrix, we considered the following alternatives:

   a. *Percentage of cumulated singular values*: Using the vector of singular values, we can sum up singular values until we reach a specific value; we suggest using 50%, 40% and 30% of the cumulated singular values.

   b. *Absolute value of cumulated singular values equals number of documents*: Here the sum of the first *k* singular values equals the number of documents in the corpus.

   c. *Percentage of number of terms*: Alternatively the number of used factors can be determined by a fraction of used terms. Typical fractions are 1/30 or 1/50.

   d. *Fixed number of dimensions*: A less sophisticated but common approach is to use a fixed number of singular values, for instance 10. For testing the other influencing factors, we chose 10 as default value.

4. *Similarity measures:* Finally, we tested three similarity measures: the Pearson-correlation, Spearman’s rho and the cosine. As default we used Spearman’s rho.

3. Reporting Results

In the pre-processing stage, stop-words removal alone (Spearman’s rho = .282) and the combination of stopping and stemming (r = .304) correlated significantly with the human scores (with a p-value less than .05). Stemming alone, however, reduced the scoring correlations.

For the weighting-schemes, the raw term frequency (tf) combined with the inverse term frequency (idf) (r = .474) as well as the logarithm (log) combined with idf (r = .392) proved best ($p < .01$). Similarly, the binary term frequency (bintf) in combination with idf ($r = .360$) showed significant results for a level of $p < .05$. Looking at the local schemes separately, we found that none of the schemes alone improved results significantly. For the global schemes, idf yielded outstanding results. Surprisingly, neither of the two schemes proposed in other literature (i.e. logarithm as the local scheme and entropy as the global) returned the expected sound results. In fact, for our case they both reduced the performance of LSA.

In our dimensionality tests, the only procedure yielding significant results was the use of a certain percentage of the cumulated singular value. On a level of $p < .01$ we received a correlation with the human grades of $r = .436$ for a share of 50 %, $r = .448$ for 40 % and $r = .407$ for 30 %. The other methods failed to show significant influence.

Finally, spearman's rho obtained the best results when comparing the influence of different similarity measures on the effectiveness of LSA. It was the only measure producing a correlation on a level of $p < .01$ with the human scores.

4. Conclusions and Future Work

Our results give evidence that for the real-world case we tested, the identified parameters influence the correlation of the machine assigned with the human scores. However, several recommendations on the adjustment of these parameters proposed in the literature do not apply in our case. We suspect that their adjustment strongly relies on the document corpus used as text base and on the essays to be assessed. Nevertheless, significant correlations between machine and human scores were discovered, which ensures that LSA can be used to automatically create valuable feedback on learning success and knowledge acquisition. Based on these first results, we intend to test the dependency of the parameter settings on each other for all possible combinations. Additionally, the stability of the results within the same discipline and in different contexts needs to be further examined. Moreover, we intend to investigate scoring of essays not against best-practice texts, but against single aspects, as this would allow us to generate a more detailed feedback on the content of essays.

References

[1] Landauer, T., Foltz, P., Laham, D. (1998): Introduction to Latent Semantic Analysis, In: Discourse Processes, 25, pp. 259-284

[2] Nakov, P., Valchanova, E., Angelova, G. (2003): Towards Deeper Understanding of the LSA Performance. In: Recent Advances in Natural language processing – RANLP'2003, pp. 311-318.

[3] Nakov, P., Popova, A., Mateev, P. (2001): Weight functions impact on LSA performance. In: Recent Advances in Natural language processing – RANLP'2001. Tzigov Chark, Bulgaria, pp. 187-193.

[4] Picciano, A. (2004): Educational Research Primer. Continuum, London.

[5] Foltz, P. (1996): Latent semantic analysis for text-based research. In: Behavior Research Methods, Instruments, and Computers, 28 (2), pp. 197-202.

[6] Foltz, P., Laham, D., Landauer, T. (1999): Automated Essay Scoring: Applications to Educational Technology. In: Proceedings of EdMedia 1999.

[7] Lemaire, B., Dessus, P. (2001): A system to assess the semantic content of student essays. In: Journal of Educational Computing Research, 24(3), pp. 303-320.

[8] Rehder, B., Schreiner, M., Laham, D., Wolfe, M., Landauer, T., Kintsch, W. (1998): Using Latent Semantic Analysis to assess knowledge: Some technical considerations. In: Discourse Processes 25, pp. 337-354.

[9] Dumais, S. (1990): Enhancing Performance in Latent Semantic Indexing (LSI) Retrieval. Technical Report, Bellcore.

[10] Berry, M., Dumais, S., O'Brien, G. (1995): Using Linear Algebra for Intelligent Information Retrieval, In: SIAM Review, Vol. 37(4), pp. 573-595.

[11] Porter, M.F. (1980): An algorithm for suffix stripping, In: Program, 14(3), pp. 130-137

[12] Hearst, M. (2000): The debate on automated essay grading, In: IEEE Intelligent Systems, 15(5), pp. 22-37

This page intentionally left blank

Young Researchers Track

This page intentionally left blank

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Argumentation-Based CSCL: How students solve controversy and relate argumentative knowledge

Marije VAN AMELSVOORT & Lisette MUNNEKE
Utrecht University, Heidelberglaan 2, 3584CS Utrecht, The Netherlands
e-mail: m.a.a.vanamelsvoort@fss.uu.nl; e.l.munneke-devries@fss.uu.nl

Our study focuses on argumentative diagrams in computer-supported collaborative argumentation-based learning. Collaboration and argumentation are crucial factors in a learning process, since they force learners to make their thoughts explicit, and listen and react to the other person's ideas. Since most people only have knowledge about part of a certain domain, argumentative interaction can help them to collaboratively acquire, refine, and restructure knowledge in order to get a broader an deeper understanding of that domain.

However, argumentative interaction is not easy. People especially have difficulties with handling controversy in arguments, and exploring their argumentative (counter)partner's ideas. An argumentative diagram might solve the above-mentioned problems by making controversy explicit, or by focusing on relations between arguments.

Thirty pairs of students discussed two cases on the topic of Genetically Modified Organisms via the computer. They communicated via chat. One third of the pairs constructed a diagram using argumentative labels to describe the boxes in the diagram. One third of the pairs constructed a diagram using argumentative labels to describe the arrows between the boxes in the diagram. The third group was asked to collaboratively write a text without using labels. We hypothesized that students who have to explicitly label arguments in a diagram will have a deeper discussion than students who do not use labels, because it helps them to focus on the deepening activities of counter-argumentation and rebuttal, and to realize what kind of argumentation they haven´t used yet. Students who have to label relations will address controversy more than students in the other two groups, because the labeling is a visual display of the controversy and might 'force' students to solve these kinds of contradictions in collaboration.

At this moment, eight pairs have been analyzed on exploration of the space of debate and labeling their diagrams. These preliminary results show that students hardly ever discuss controversy and relations in chat, nor talk about the labeling of the diagram. They are mainly focused on finishing the diagram or text, without explicitly exploring the space of debate together. They seem to avoid controversy, probably because they value their social relation, and because they want to finish the task quickly and easily.

Students mainly explore the space of debate in the diagrams. The diagrams in the label-arrow condition are bigger than the diagrams in the label-box condition. There was no difference in conditions in amount of counterarguing or rebutting arguments in the diagram. Most students indicated there was no controversy in their discussion with their partner. However, when looking at the diagrams, many controversies can be found that are not related or discussed. We wonder whether students do not see controversy or whether they don't feel the need to solve it. Further results will be discussed at our presentation.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Generating Reports of Graphical Modelling Processes for Authoring and Presentation

Lars BOLLEN
University of Duisburg-Essen, Faculty of Engineering
Institute for Computer Science and Interactive Systems, 47048 Duisburg, Germany

In the process of computer supported modelling, the learner interacts with computational objects, manipulates them and thereby make his thoughts explicit. In this context, the phrase "objects to think/work with" has been introduced in [1], meaning that the exploration, manipulation and creation of artefacts support in establishing understanding.

Nevertheless, when a learner finishes a modelling task within a modelling environment like Cool Modes [2], usually only a result is stored. The process of creating and exploring a model is compressed to a single artefact. Information about *the process* of his work, about *different phases,* about the *design rationale, alternative solutions* and about *collaboration* gets lost when having only a single artefact as the output of a modelling process.

Knowledge about these issues is helpful for various target groups and for various purposes: E.g., the learner could use this information for self reflection, peer authoring and for presenting own results. Teachers could be supported in assessment, authoring and in finding typical problems in students solutions. Researchers in the field of AIED and CSCL could use the additional information for interpreting and understanding learner's actions.

Approaches that take into account processual aspects of learning and modelling can be found in [3, 4]. The problem described above can be addressed and solved by generating *reports*. Reports, in the sense of this approach, are summaries of states and action traces from modelling processes. A prototypical implementation of a report generation tool is already available. In this implementation, information about states and action traces from modelling processes are collected, analysed (using domain knowledge) and represented automatically in a graph-based visualisation, in which different nodes represent different states of the modelling process. Edges represent the actions that led to these states, providing information for analysing and interpreting modelling processes. Combining this automatic generated, graph-based representations with a mechanism for feeding back states into the learning support environment, provides for *authoring* and *presentations* (playing back previously recorded material)*, monitoring* and *assessment* (observing collected material) and *research* (using advanced analysis methods to inspect specific features of modelling and collaboration).

References

[1] Harel, I. and Papert, S. (eds.) (1991): *Constructionism.* Ablex Publishing. Norwood, NJ.

[2] Pinkwart, N. (2003). *A Plug-In Architecture for Graph Based Collaborative Modelling Systems.* In Proc. of the 11th Conference on Artificial Intelligence in Education (AIED 2003), Amsterdam, IOS Press.

[3] Müller, R., Ottmann, T. (2000). *The "Authoring on the Fly" System for Automated Recording and Replay of (Tele)presentations.* Special Issue of Multimedia Systems Journal, Vol. 8, No. 3, ACM/Springer.

[4] Koedinger, K. R., Aleven, V., Heffernan, N., McLaren, B. M., and Hockenberry, M. (2004). *Opening the Door to Non-Programmers: Authoring Intelligent Tutor Behavior by Demonstration.* In Proceedings of 7th International Conference on Intelligent Tutoring Systems, ITS 2004, Maceio, Brazil.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Towards An Intelligent Tool To Foster Collaboration In Distributed Pair Programming

Edgar **ACOSTA CHAPARRO**
IDEAS Lab, Dept. Informatics, University of Sussex BN1 9QH, UK
E.A.Chaparro@sussex.ac.uk

Pair programming is a novel, well-accredited approach to teaching programming. In pair programming (as in any other collaborative learning situations) there is a need for tools that support peer collaboration Moreover, we must bear in mind the strong movement towards distributed learning technologies and how this movement could influence the design of such tools[1]. Indeed, there have been some attempts to implement tools to support distributed pair programming [2]. However, none of them have had any influence of pedagogical theories. To support the design and implementation of an intelligent tool in this work, the Task Sharing Framework (TSF) developed by Pearce et al. [3] is being explored.

The aim of this doctoral research is to investigate the suitability of the TSF [3] in the design and implementation of a prototype of an intelligent tool that monitors and enhances the collaboration between distributed pair programmers facilitating their efforts at learning programming. In particular, the tool will search for signs of collaboration difficulties and breakdowns of pair programmers solving exercises of object-oriented programming.

The TSF will support the sharing of collaborative tasks between users. Each peer will have their own identical yet independent copy of the task that by default, only they themselves can manipulate. The visual representation of agreement and disagreement has the potential to constructively mediate the resolution of collaborative disputes [3]. Programming is a heavy cognitive task and with the TSF each student will have two representations to look at. This might impact students' cognitive efforts.

The author is interested in exploring the learning gains and the peer collaboration with different versions of the intelligent tool using the TSF. Each participant will do a pre-test to evaluate her level of expertise in object-oriented programming. The learning gain and the collaboration will be measured by comparing the results from pre and post-tests, plus by analysing verbalizations and performance on the task. If the intelligent tool can be established and the TSF prove to be effective, it will support the implementation of intelligent tools that will extend the benefits of pair programming to a large population. Progress in this would also be of major significance in the area of intelligent learning systems used for teaching programming.

References

1. Fjuk, A., *Computer Support for Distributed Collaborative Learning. Exploring a Complex Problem Area.*, in *Department Informatics - Faculty of Mathematics and Natural Sciences.* 1998, University of Oslo: Olso. p. 256.
2. Stotts, P.D. and L. Williams, *A Video-Enhanced Environment for Distributed Extreme Programming.* 2002, Department of Computer Science. University of North Carolina.
3. Pearce, D., et al., *The task sharing framework for collaboration and meta-collaboration*, in *(in press) proceeding of 12th International Conference on Artificial Intelligence in Education.* 2005: Amsterdam - Netherlands.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Online Discussion Processes: How do earlier messages affect evaluations, knowledge contents, social cues and responsiveness of current message?

Gaowei Chen
Department of Educational Psychology,
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong

This study examined how earlier messages affected the four properties of current message, i.e., evaluations, knowledge contents, social cues and responsiveness. If earlier messages help to explain these features in current one, we can further know the interrelationship of online messages, and thereby taking measures to improve online discussion processes. Most current studies focused on dependent forums, which are related to specific courses, to do content analysis of online discussion. This study extended this line of research by examining how online discussion messages affect one another in an independent academic discussion forum.

I selected 7 hot topics from the math board, an academic discussion forum of the Bulletin Board System (BBS) Website of Peking University (http://bbs.pku.edu.cn). This independent forum is free for entrance or leaving, with little requirement or limitation for participants' activities. There were totally 131 messages, 47 participants responding to the 7 topics. After coding data, I did regressions at the message level. Structural equation model (SEM) was also used to test direct and indirect effects in the analyses.

Results showed that, disagreement and contribution in previous message positively predicted disagreement and personal feeling in current message. Visit number of previous poster was likely to increase contribution in current message, while personal feeling in message two turns prior tended to weaken it. Disagreement in current message raised the likelihood of it getting future response. Moreover, replying to a previous on-topic message can also help the current message to draw later response. Together, these results suggest that evaluations, knowledge contents, social cues and person status in earlier messages may influence the property of current message during online discussion processes.

Further studies are necessary before making firm recommendations. However, results of this study suggest that designers and teachers may improve the quality of online academic discussion by taking the following advices.

Attach more earlier messages to current message. The branch structure of online discussion made it difficult for current poster to track earlier messages. As shown in the results and discussion, only lag 1 and lag 2 messages, which were displayed together, can affect current message. To help participants understand the discussion thread more easily, designers can attach more earlier messages to current post, e.g., adding lag 3 and lag 4 messages. Some BBS websites have adopted this kind of discussion style, e.g., the "unknown space" BBS website (http://www.mitbbs.com).

Carry on controversial discussion in online forum. As shown in this study, participants were likely to perform and continue controversial interactions in online discussion. It implies that teachers can move some controversial topics, e.g., new theories or problems without certain answers, to online forum for discussion. Under such topics, participants can easily come into different sides to controvert and argue by posting personal ideas.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

PECA: Pedagogical Embodied Conversational Agents in Mixed Reality Learning Environments

Jayfus T. Doswell
George Mason University

The Pedagogical Embodied Conversational Agent (PECA) is an "artificially intelligent", computer 3D graphic, animated character that teaches from computer simulated environments and naturally interacts with human end-users. What distinguishes a PECA from the traditional virtual instructor or pedagogical agent is the PECA's ability to intelligently use its 3D graphical form and multimodal perceptual ability. While so doing, the PECA has capabilities to communicate with human end users and demonstrate a wide variety of concepts from within interactive mixed reality environments. More importantly, the PECA uses this intuitive form of communication to deliver personalized instruction for enhancing human learning performance by applying its underlying knowledge of empirically evaluated pedagogical techniques and learning theories. A PECA combines this "art and science" of instruction with knowledge of domain based facts, culture, and an individual's learning strengths in order to facilitate a more personal human learning experience and to improve its own instructional capabilities. The challenge, however, is engineering a realistically behaving 3D character for human interaction in computer simulated environments and with capabilities to provide tailored instruction based on well defined pedagogical rules and knowledge of human learning capabilities across cultures.

Neither the PECA's advanced human computer interface capabilities or ability to interact within mixed reality environments is useful without it's knowledge of best instructional methods for improving human learning. A formal instructional method is called **pedagogy** and is defined as the *art and science of teaching*. PECA pedagogy may include *scaffolding* techniques to guide learners when necessary; *multi-sensory* techniques so students use more than one sense while learning; multi-cultural awareness where awareness of the individual's social norms potentially influences learning outcomes, among other instructional techniques. The PECA also tailor a particular instructional method to, at minimum, weighted learning strengths, including: *visual learning* seeing what you learn; *auditory learning* hearing spoken messages or sounds to facilitate learning; *kinesthetic learning* to sense the position and movement of what is being learned; and *tactile* learning where learning involves touch. These pedagogical and learning styles may be structured and decomposed, without losing their inherent value, into a 'codifed' set of computational rules expressed, naturally, by the PECA.

This paper presents a novel approach to building PECAs for use in mix reality environments and addresses key challenges researchers face in integrating pedagogy and learning theory knowledge in PECA systems.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Observational Learning from Social Model Agents: Examining the Inherent Processes

Suzanne J. EBBERS and Amy L. BAYLOR
Centre for Research in Interactive Technologies for Learning (RITL)
Learning Systems Institute, Florida State University, Tallahassee, FL 32306

Using computers as social information conveyors has drawn widespread attention from the research world. Recently, the use of pedagogical agents has come to the forefront of research in the educational community. Already they are termed “social interfaces”. Yet for them to be fully useful, we must delineate how similarly to humans they socially function.

Researchers are looking at them as social models. It would be useful to examine human-human modeling studies and replicate them in using agents. Schunk & colleagues [1-2] studied Mastery and Coping models in a social learning situation and their impact on self-efficacy, skill, and persistence. These model types have not been researched using agents.

Social interaction with agents is another activity whose social impact has not much been examined. In human-human social learning situations, interaction with a model is more intensely experienced than is a vicariously experience. No study has compared the impact of directly or vicariously experienced social interaction by humans with pedagogical agents.

Threat creates dissonance. We affiliate to reduce dissonance. Under threat one would seek to affiliate with a similar other. If the only “other” available is an agent, learners should seek to affiliate depending on agent similarity features. If the “similar” Mastery model demonstrates non-threatened learning through cheerful self-efficacy while the “similar” Coping agent demonstrates a threatened experience through initial self-doubt and apprehension, then learners should disaffiliate from the Mastery agent and affiliate with the Coping model. Direct social interaction will intensify learning efforts.

The primary purpose of the 2x2 factorial design research is to examine the impact of social model agent type (Mastery, Coping) and social interaction type (Vicarious or Direct) on participant motivation (self-efficacy, satisfaction), skill, evaluations, frustration, similarity perceptions, attitude and feelings about experience. Secondarily, the study will use descriptive statistics describing how social processes manifest in affiliation activities.

The computerized instructional module teaches learners to create an E-Learning-based instruction plan. A “teacher” agent provides information. The agent “listens” to the “teacher” except when self-expressing to a “classmate” agent or the learner, who then responds. Participants will be about 100 university pre-service teachers in an intro tech class. The experiment will occur during a class 1.5 hour session. The participants will be randomly assigned to one of the five conditions (including control – no agent present). Analysis will consist of two-way ANOVAs on most variables. For Motivation a two-way MANOVA will be used. “Feelings” will be qualitatively analyzed.

References

[1] D. H. Schunk and A. R. Hanson, "Peer-Models: Influence on Children's Self-Efficacy and Achievement," *Journal of Educational Psychology*, vol. 77, pp. 313-322, 1985.

[2] D. H. Schunk, A. R. Hanson, and P. D. Cox, "Peer-Model Attributes and Children's Achievement Behaviors," *Journal of Educational Psychology*, vol. 79, pp. 54-61, 1987.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

An Exploration of a Visual Representation for Interactive Narrative in an Adventure Authoring Tool

Seth GOOLNIK
The University of Edinburgh

Research Summary

The earlier Ghostwriter project attempted to address the issue of weaknesses in children's writing skills through the development of a virtual learning environment targeted to improve them. Ghostwriter is a 3D interactive audio-visual adventure game, in using it, results showed that children found this experience to be highly motivating and stories written after use of the software displayed significantly better characterisation than those written in typical classroom conditions.

The work of Yasmin Kafai suggests that improved learning can be obtained by allowing children to create learning environments themselves. Motivated by this the Adventure Author project aims to explore if, by developing an authoring tool to allow children to not only participate in interactive narrative environments *à la* Ghostwriter but in addition enable them to create these narratives themselves, it would be possible to capitalise on the benefits of the Ghostwriter.

As a continuation of Adventure Author this project attempted to formalize a system for visually representing interactive narrative as the next logical step in the development of a 3D virtual environment authoring tool. It then investigated whether children of the target age range for the authoring tool could understand and generate interactive narratives using this representation, attempting to provide a solid foundation for the ultimate development of the authoring tool.

The visual system was developed using the example interactive narrative of adventure game books, with this found to be formalizable within the representational structure of an Augmented Transition Network. This system was first presented to the children via a specially designed interactive narrative, structurally contained on a paper chart. After participating in the interactive story the children were able to understand as a group that the chart represented it and further they were able to fully generate their own interactive narrative using the same paper-based representation.

Following the success of the paper-based medium in conveying the visual system the computer-based medium of AA2D was developed. In individually using AA2D to both understand and generate the representation of interactive narrative all participants were successful: all understood the formal system AA2D conveyed; and all were able to use AA2D to generate their own valid interactive narratives. Participants also all explicitly commented they had enjoyed using AA2D for these purposes and would be happy to do so again.

This project thus provides a clear assertion that the potentially valuable Adventure Author project can and should continue. By developing a visual formalisation of interactive narrative and then demonstrating that children of the target age range can both understand and generate it, an ultimate 3D interactive narrative environment authoring tool can now be seen to be viable. Furthermore, given that all experimental participants were admittedly engaged by their experiences and that surveyed literature suggests the educational benefits of their production, this project has shown that such further exploration into interactive narrative through virtual environments has real educational potential.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Affective Behavior in Intelligent Tutoring Systems for Virtual Laboratories

Yasmín HERNÁNDEZ[1], Julieta NOGUEZ[2]
[1] *Gerencia de Sistemas Informáticos, Instituto de Investigaciones Eléctricas*
myhp@iie.org.mx
[2] *Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Cd. de México*
jnoguez@itesm.mx
México

We are developing an intelligent tutoring system coupled to a virtual laboratory for teaching mobile robotics. Our main hypothesis is that if the tutor recognizes the student affective state and responds accordingly, it may be able to motivate the student and improve the learning process. Therefore, we include in the ITS architecture an affective student model and an affective behavior model for the tutor.

The student model contains knowledge about the affective state of the student. Based on the OCC model [1], we establish the affective state as an appraisal between goals and situation. To determine the student affective state we use the following factors: student personality traits, student knowledge state, mood, goals and tutorial situation (i.e. outcome of the students' actions). According to the OCC model, the goals are fundamental to determine the affective state; we infer them by means of personality traits and the cognitive student state. For the personality traits we use the Five Factor Model [2] which considers five dimensions for personality. We use three of them to establish goals, because these are the ones that have more influence on learning. We represent the affective student model by a Bayesian network; since this formalism provides an effective way to represent and manage the uncertainty inherent in student modeling [3].

Once the affective student model has been obtained, the tutor has to respond accordingly and to provide the student with a pedagogical response that fits with his affective and cognitive state. The affective behavior model (ABM) receives information from the affective student model, the cognitive student model and the tutorial situation; and translates them into affective actions for the tutor and interface modules. The affective action includes knowledge about the overall situation that will help the tutor module to determine the best pedagogical response to the student, and also will advise the interface module to express the response in a suitable way. We represent the ABM by means of a decision network, where the affective action considers utilities in learning and motivation.

Currently, we are implementing the affective student model and integrating it to the cognitive student model. We are preparing some experiments and looking for pedagogical and psychological support for the formalization of the affective behavior model.

References

[1] Ortony, A., Clore G.L., and Collins A., *The Cognitive Structure of Emotions*, Cambridge University Press, 1988.

[2] Costa, P.T. and McCrae, R.R., *Four Ways Five Factors are Basic*, Personality and Individual Differences, 1992, 13 (1), pp. 653-665.

[3] Conati, C., and Zhou X., *Modeling students' emotions from Cognitive Appraisal in Educational Games*, 6th International Conference on Intelligent Tutoring Systems, ITS 2002, Biarritz, France, pp. 944-954.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Taking into account the variability of the knowledge structure in Bayesian student models.

Mathieu HIBOU
Crip5 Université René Descartes – Paris 5
45 rue des Saints-Pères 75270 Paris Cedex 06 France
mathieu.hibou@math-info.univ-paris5.fr

Abstract. Bayesian belief networks have been widely used in student and user modelling. Their construction is the main difficulty for their use in student modelling. The choices made about their structures (especially the arcs orientation) have consequences in terms of information circulation.
The analysis we present here is that the network structure depends on the expertise level of the student. Consequently, the evolution of the network should not only be numerical (update of the probabilities) but also structural. Hence, we propose a model constituted of different networks in order to take into account these evolutions.

Bayesian networks (BN) have been successfully used for student modelling in many different systems, [1], [4], [5]. We propose to extend their use in order to take into account the changes in the student's knowledge structure. The existence of structural differences between experts and novices knowledge and problems representations have been studied and highlighted in cognitive psychology [3]. Consequently, there should be an evolution not only of the network parameters but also of its structure to reflect the changes in the student's knowledge structure.
The solution we propose to take into account these changes, inspired by the Bayesian learning approach [2], is to consider that the model is constituted of different sub-models, each one of them being a Bayesian network. The selection of the most appropriate sub-model is made using abductive inference. After observation, the most probable explanation is figured out for each network, $v_i^{abd} = \arg\max_{v \in V \setminus e}\left(P\left(V = v \middle| e\right)\right)$, where i denotes the network and e the evidence observed. Each of those explanations has a probability $P\left(V = v_i^{abd} \middle| e\right)$, and this probability is the criteria used for the determination of the sub-model that fits the best.
This idea is currently tested in order to determine whether or not we can detect different sub-models.

References

[1] A. Bunt, C. Conati. Probabilistic student modelling to improve exploratory behaviour, *in Journal of User Modeling and User-Adapted Interaction*, volume 13 (3), pages 269-309, 2003.
[2] W. L. Buntine. Operations for learning with graphical models, *Journal of Artificial Intelligence Research*, volume2, n°, pages159-225, 1994.
[3] Chi, M.T.H., Feltovitch, P.J., & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science, 5,* 121-152.
[4] C. Conati, A. Gertner, K. Vanlehn. Using Bayesian networks to manage uncertainty in student modeling, in *Journal of User Modeling and User-Adapted Interaction*, volume 12 (4), pages371-417, 2002.
[5] A. Jameson. Numerical uncertainty management in user and student modeling: an overview of systems and issues, in *User- Adapted Interaction*, volume 5 (3-4), n°5, pages193-251, 1996.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Subsymbolic User Modeling in Adaptive Hypermedia

Katja HOFMANN
California State University, East Bay, 25800 Carlos Bee Blvd., Hayward, CA 94542, USA,
Phone +1(510) 885-7559, E-mail khofmann@horizon.csuhayward.edu

The most frequently used approach to user modeling in adaptive hypermedia is the use of **symbolic machine learning techniques**. Sison and Shimura and Weber and Brusilovsky describe a number of current systems, which use for example decision trees, probabilistic learning, or case-based reasoning to infer information about the student. However, many researchers have come to the conclusion that the applicability of symbolic machine learning to user modeling in adaptive learning systems is inherently limited because these techniques do not perform well on noisy, incomplete, or ambiguous data. It is very hard to infer information about the user based on the observation of single actions.

Neural networks and fuzzy systems are **subsymbolic machine learning techniques** and are a very promising approach to deal with the characteristics of data obtained from observing user behavior. The two techniques complement each other and have inherent characteristics that make them suitable to deal with incomplete and noisy data inherent to user behavior in hypermedia systems. Most importantly, this approach can identify similarities in underlying patterns of complex, high-dimensional data.

I want to find out how subsymbolic machine learning can be used to adapt navigation of web-based tutorials to the goals, knowledge level, and learning style of the student. The students' interaction with the tutorial will be recorded and form the input to a neuro-fuzzy clustering mechanism. The resulting clustering will group similar student behavior in clusters, which is a representation of the patterns underlying the user behavior. My **hypothesis** is that students with similar goals, background knowledge, and learning style will show similar user behavior and will thus be grouped in the same or adjacent clusters. Based on the clustering, the online tutorial will adapt the navigation by placing the documents that similar students found helpful in the most prominent position.

My work is based on the existing ACUT tutorial. ACUT uses collaborative learning and social navigation and aims at increasing retention of Computer Science students without extensive knowledge on UNIX, especially women and minority students.

After implementing the clustering mechanism I will use **empirical evaluation** to test my hypothesis. Focused interviews will be used to receive very detailed qualitative and quantitative data. The Results will give information about the effectiveness and applicability of the adaptation mechanism, and about the evaluation method.

The presented research is a work in progress and **future research** will be needed to carefully evaluate and compare the efficiency of current technologies and subsymbolic clustering for user modeling in adaptive hypermedia systems. After evaluating the first results I will be able to analyze resulting clustering and recommendations and refine the algorithm to make more informed decisions about navigational adaptation.

The results of this research will be applicable to user modeling, navigation design, and development of collaborative computer based learning systems and recommender systems.

Acknowledgements. I want to thank my advisor Dr. Hilary J. Holz for her invaluable help, feedback and motivation. I also thank Dr. Catherine Reed for her help with the educational side of my research. This work is partially sponsored by an ASI fellowship of CSU EB.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

The Effect of Multimedia Design Elements on Learning Outcomes in Pedagogical Agent Research: A Meta-Analysis

Soyoung Kim
Instructional Systems Program
RITL – PALS
http://ritl.fsu.edu
Florida State University
syk02c@fsu.edu

This study aimed at synthesizing the results of experimental research on the effect of multimedia elements in pedagogical agents on learning outcomes by using a meta-analysis technique. This pilot study targeted the overall effects of treatments that varied according to design elements and learning outcomes. Furthermore, the results of this meta-analysis were expected to provide in-depth understanding about pedagogical agent research in a more systematic way.

Previous research suggests that lifelike agents have a strong motivational effect, promote learners' cognitive engagement, and arouse various affective responses. However, the results of research on pedagogical agents are somewhat varied across studies due to the nature of the embryo stage. This study intended to explain the overall effect of multimedia elements across studies on pedagogical agents and to try to find a consensus regarding the role of multimedia elements in the effectiveness of pedagogical agents.

Twelve different experimental studies of pedagogical agents by five different authors were included in this meta-analysis, through the process of inclusion and exclusion. Unpublished manuscripts as well as published articles were incorporate for this analysis to avoid publication bias. Non-significant results as well as significant results were incorporated as long as appropriate descriptive data were reported to avoid selection bias.

Through the coding process, the four main elements of multimedia design were identified as 'treatment' variable; the three main learning outcomes were identified as 'outcome' variable. The treatment variable was classified into four different levels; (1) auditory, (2) visual image, (3) visual image plus animation, (4) visual image plus social meaning (role, gender, ethnicity, etc.). The outcome variable was categorized as (1) affective outcome, (2) cognitive outcome and (3) motivational outcome.

The key to meta-analysis is defining an effect size statistic capable of representing the quantitative findings of a set of research studies in a standardized form. A total of 28 different effect sizes from 12 different studies were obtained and incorporated in this data set.

A categorical fixed model, which is analogue to ANOVA model, was applied and a total of five different predictors including moderate variables (author group, duration and subject matter) as well as main variables (treatment, outcome) were investigated.

Results in this study indicated that the presence of a pedagogical agent transmitted the effect of multimedia design elements (Q_{total}), which were created by technological support consistently across the studies, on learning outcomes, even though the effect of each variable ($Q_{between}$) could not be verified.

Discussion focused on pedagogical agents in the context of the reciprocal relationship between learning theory and multimedia design and its impact on learning outcomes. Results suggested possible factors and, most of all, it has improved the understanding of the pedagogical agent research. Furthermore, larger size sample should be required for a better meta-analysis. In addition, more studies about affective domains should be incorporated.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

An ITS that provides positive feedback for Beginning Violin Students

Orla LAHART
School of Informatics, National College of Ireland
olahart@ncirl.ie

Abstract: Feedback is highly important within any learning environment. Providing feedback in a manner which enhances, rather than damages students self-esteem is an important skill that is seldom taught. The art of good feedback is highly complex and this complexity is further heightened when the domain is music. Beginning violin students are faced with a steep learning curve due to the complex nature of the instrument and they may find it difficult to believe that mastery of the instrument is possible. Shinichi Suzuki believed that talent is a product of environment rather than heredity and therefore mastery is possible. The research proposed in this paper involves the development of an Intelligent Tutoring System that provides an individualised positive learning environment for beginning violin students practicing at home. The pedagogical framework that informs the system is the Suzuki method as it is based on the premise that through positive feedback students can reach their potential.

The research proposed in this paper is concerned with the development of an Intelligent Tutoring System (ITS) that uses the Suzuki method to inform its pedagogical framework. The system monitors the student, infers a student model and delivers feedback on the basis of explicit teaching strategies. Current research is informative in terms of diagnosis and teaching strategies [1, 2]. However, little research focuses on music preformance and positive feedback.

The proposed architecture consists of three modules. The Student Assessment model monitors student's performance. The Student model captures information on student progress. The Adaptive Decision model makes intelligent recommendations based on the Student model.

The Suzuki Method is based on the premise that talent is a product of environment rather than heredity [3]. The keys to the success of the Suzuki Method include motivation, repetition, listening, positive reinforcement, and parental involvement. The ITS proposed here will harness these concepts, in particular positive feedback, with the aim of encouraging beginning violin students to reach their potential.

[1] Brandão, M., Wiggins, G. & Pain, H. (1999). Computers in Music Education. *Proceedings of the AISB'99 Symposium on Musical Creativity*.

[2] Dannenberg, R., Sanchez, M., Joseph, A., Capell, P., Joseph, R. And Saul, R. (1990). A Computer-Based Multi-Media Tutor for Beginning Piano Students. *Interface – Jounal of New Music Research*, 19(2-3), pp.155-173.

[3] Suzuki, S. (1986). Nurtured by Love: The Classic Approach to Talent Education 2nd edition. Suzuki Method International.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

A Proposal of Evaluation Framework for Higher Education

Xizhi Li
The CKC honors School of
Zhejiang University, P.R. China

Hao Lin
The CKC honors School of
Zhejiang University, P.R. China

Summary. In traditional education evaluation system, the goal is looking for appropriate methods (mostly numerical) to measure individual performance. While these highly compact measuring results provide some insights to the individual itself; they are less useful to other people and should avoid being used for public comparisons. An inadequate or unnecessary evaluation to the individual performance will usually lead to negative impact on the individual's learning and working experiences in a social environment. To summarize, the traditional evaluation system takes a synthetic or regressional approach in general. By contrast, we propose an opposite approach, in which the focus of evaluation shifts from information regression to information aggregation (see Figure 1). The philosophy behind this approach is to provide as much information about the individual as possible and reveal it in an ease-to-access manner to the individual as well as people who are interested in them. In the information age, it is technically possible to adopt information-rich evaluation system. We also believe that the trend of education evaluation will be the emergence of various intermediate (intelligence) technologies and frameworks which make this new approach dominant in all education systems.

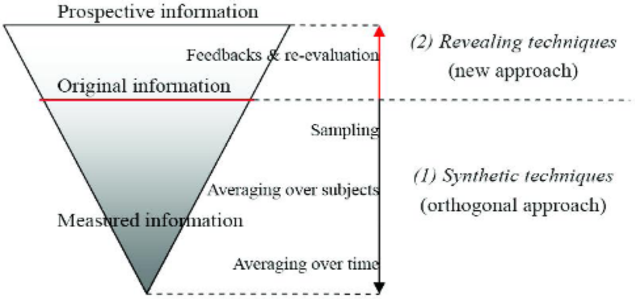


Figure 1. Two directions of evaluation techniques

As part of our proposal, we designed a web agent based evaluation framework. It demonstrates one possible way to incorporate the new evaluation philosophy into the existing education evaluation system. In the framework: from the individual's point of view, evaluation is a voluntary act to publicize its finished or ongoing works, ideas, or even plans; from the community's point of view, these publicized artifacts automatically enjoy the right to be evaluated and referenced. Hence, evaluation becomes an open process of information aggregation with annotated feedbacks, formative reviews, cross references and re-evaluation. This aggregated evaluation database (or autobiography) will become an important product that both the individual and the education institute jointly deliver to the society and might continue to be useful after the individual starts his or her professional career.

The learner model for undergraduate level college student is also studied, on which the proposal is grounded.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Supporting collaborative medical decision-making in a Computer-based learning environment

Jingyan Lu
Dept. of Educational and Counselling Psychology, McGill University
3700 McTavish Street, Montreal, Canada, QC H3A 1Y2
Email: jingyan.lu@mail.mcgill.ca

Objectives of the study

This proposed research is to design a theoretically driven and pedagogically grounded computer-based learning environment (CBLE) to generate naturalistically compelling emergency scenarios requiring medical students to collaborate in decision-making and knowledge building. I will investigate how students' decision-making activities, and thereby their conceptual understandings, change as a function of the collaborative learning processes. I will look at how individual cognition affects social interaction and how social interaction affects individual cognition. Furthermore, I will examine how CBLE shapes individual and collaborative learning.

Background of the study

The design of CBLE is driven by the increasing interests in collaborative learning theories. Two important cognitive processes, elaboration and co-construction, which have been found to lead to deep understanding [1, 2], are integrated in CBLE. It is pedagogically grounded because it takes the consideration of the characteristics of medical emergencies which demand efficient deployment of relevant knowledge, experience and skills in dynamic, highly stressful contexts. In addition, it is empirically interesting because it is based on two years observation and investigation on clinical teaching of medical decision-making in emergency medicine.

Design Framework

Two activities, dynamic decision-making and structured synchronous discussion are supported in CBLE. The former is a role-playing decision-making activity between the medical students and the teacher. Students play the role as the doctor and receive dynamic feedback from the teacher who plays the role as the patient and the nurse. The latter is the synchronous discussion structured based on emergency algorithm. The CBLE will incorporate visualization tools and argumentation tools to support both activities in a naturally distributed learning environment.

1. Schmidt, H.G., et al., *Explanatory models in the processing of science text: the role of prior knowledge activation through small-group discussion.* Journal of Educational Psychology, 1989. **81**: p. 610-619.
2. Savery, J.R. and T.M. Duffy, *Problem based learning: An instructional model and its constructivist framework.* Educational Technology, 1995. **35**(5): p. 31-37.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Logging, Replaying and Analysing Students' Interactions in a web-based ILE to Improve Student Modelling

Manolis Mavrikis [a,b,1],
[a] *School of Mathematics, The University of Edinburgh*
[b] *School of Informatics, The University of Edinburgh*

1. Summary

With the advancement of computers in education and the emergence of e-learning, interactive learning environments are becoming more and more integrated in the classroom. This provides an opportunity to conduct research under more genuine situations. For example, studies can be linked (if carefully designed) to the usual material students have to learn or the assessments that they have to take. In addition, more and more students perceive this medium differently and are getting used to its role in the classroom. Studies in the same medium that students are using for their learning yield more realistic and accurate data and can help in fine-tuning the system more successfully.

Although it is true that aspects of student modelling can be investigated outside the specific learning situation, it is quite difficult to separate the confounding factors or avoid Hawthorn-like effects especially when one is interested in affective states and traits of the students. Therefore, by conducting research in as realist conditions as possible we can achieve a better understanding of the learning process, of the students' behaviour and the actions that make sense in this particular context. One of the assumption underlying this research is that students interact differently when they are working alone than in a situation where the tutor's presence influences their behaviour. For instance, we have already established that students are misusing hint and help facilities, easily quit concepts and exercises in which they are not interested and, in general, exchibit a different behaviour than when interacting with a human tutor.

This paper explains further the rational for collecting fine-grained student's interaction and briefly presents the technology behind the agent for logging them. Based on the assumption that despite the low bandwidth of information, there are some subtle aspects of this interaction that can be taken into account to improve diagnosis, the interactions are replayed to experts to elicit diagnostic rules and knowledge about the actions the system should take. Here, preliminary results from pilot studies and ways of visualising the interactions are presented. Thes can help determining the best level of abstraction, useful moments for replaying, and ways of conducting future computational analyses.

[1] Correspondence: JCMB, Mayfield Road, Edinburgh EH9 3JZ, UK. Tel.: 0-131-6505076; E-mail: m.mavrikis@ed.ac.uk.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

How do Features of an Intelligent Learning Environment Influence Motivation? A Qualitative Modelling Approach

Jutima METHANEETHORN[1]
The SCRE Centre,University of Glasgow, Glasgow, UK, G3 6NH
jutima@scre.ac.uk

Recent research points to the notion that motivation is a crucial factor when creating Intelligent Learning Environments (ILEs). Yet the research in motivation in tutoring systems has not considered relationships between features of ILEs and components of learners' motivational structure. Several interesting questions can be addressed: (1) How do these features impact on the motivational structure of learners? (2) What is the evidence that cause-effect relationships exist between those features and the components of learners' motivation? (3) If so, what is the nature of these relationships? This paper proposes to use a qualitative modelling approach [1] to model motivational characteristics of learners while interacting with an ILE, particularly within the context of narrative-based educational game. The motivation for applying this approach stems from our consideration of motivation as a dynamic and complex system which is difficult to inspect. A preliminary causal model that shows the relationship between a learner's motivation and ILE features was created. We also employ qualitative process theory [2] to define a simple notion of process, the ways in which things change over time, among elements in our motivation model because we suspect that there is some sort of function associated with it. We are applying our model in the context of role-playing games (RPGs) in which human players assume the characteristics of some person or creature type. The domain knowledge that we aim to teach is the concept of Entity Relationship Modelling (ERM). The future steps of our research include not only the development of the model simulation, but also validation for its plausibility. We plan to deploy our system with a group of students to estimate values of their motivation. The data collected from the experiment will be used to compare with the model's behaviour to see if the model needs to be changed to make it more consistent.

References

1. Brown, J.S. and J. de Kleer. *A framework for a qualitative physics*. in *the Sixth Annual Conference of the Cognitive Science Society*. 1984.
2. Forbus, K.D., *Qualitative Process Theory.* Artificial Intelligence, 1984. **24**: p. 85-168.

[1] The author is a PhD student under the supervision of Prof. Paul Brna. Many thanks to him for his advice during the writing of this paper.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Integrating an Affective Framework into Intelligent Tutoring Systems

Mohd Zaliman YUSOFF
IDEAS lab, Department of Informatics, School of Science &Technology,
University of Sussex, UK
m.z.yusoff@sussex.ac.uk

Summary

This paper presents the integration of an affective framework into an Intelligent Tutoring System (ITS). This framework extends current affective learning frameworks by introducing a two layered appraisal and reaction process. The objective of the appraisal phase is to assess the students' emotional state. As to help students manage their emotions students' emotional state, both domain-dependent and domain-independent strategies and activities are used in the reaction phase of this framework.

The implementations of these layers are undertaken at two learning stages: at the beginning of a lesson and during the lesson. The primary appraisal establishes the student's emotional state with regard to his personal beliefs and goal commitments. However, the primary appraisal is envisaged only if the student feels it is necessary. The secondary appraisal uses the student's reactions to two eliciting factors to appraise emotion. These eliciting factors are: the difficulty level of the lesson which is based on the nature of the lesson and the student's control over the lesson.

As for the reaction phase, both domain- dependent and domain-independent strategies are used as a means to help students manage their emotional state. Domain-dependent strategies help students by providing suitable suggestions and tips that are adapted to the students' elicited emotional state. In comparison, domain-independent strategies include the use of coping statements and relaxation exercises. For example statements such as "I can make things happen" are used to maintain students' happiness while statements like "I can see this problem from another perspective to make it seem more bearable" are used to reduce students' nervousness. Apart from coping statements, relaxation activities such as muscle and head exercises are employed to help students manage their emotions.

Preliminary empirical work supports the hypothesis that students believe that emotions are important to leaning. In addition, the use of both domain-dependent and domain-independent strategies is perceived to be equally useful to help students manage their affective state while learning.

Keywords: emotionally sound affective framework, emotion, domain-dependent, domain-independent.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Relation-based heuristic diffusion framework for LOM generation

Olivier Motelet [1]
DCC - Universidad de Chile

Keywords. learning material reuse, learning object metadata, metadata generation

The theoretical advantage of LOM documents for reusing learning material is limited by the difficulty to generate them. Motivated by this issue, this work introduced an original method for metadata generation based on relations between LOM documents. The instructor designs the course graph specifying the relations between learning objects. Then, the system based on an extensible set of heuristics generates relevant information for LOM attribute instantiation. For example, consider a learning object l. This document contains n learning objects li (with $i \in \{1..n\}$) of smaller granularity than l. According to the LOM specification, l is related to the lis with relations of type hasPart. Moreover, the LOM attribute keyword of l should contain the values of the attribute keyword of the lis. This statement can be formulated with an acquisition heuristic. An acquisition heuristic is a formula rationalizing **the influence of** the values vi of a LOM attribute a of a set of learning objects li **on** the value v of the same LOM attribute a of a leaning object l related with a relation of type r to the lis. According to this definition, an acquisition heuristic for the attribute keyword and the relation hasPart should define that the value v contains the union of vis. A diffusion framework is in charge of applying the heuristics on the LOM attributes. The framework enables the propagation of heuristic effects by recursion. Basically, the heuristics are applied *not only* on the original values of the LOM attributes of the related learning objects, *but also* on the results of processing the diffusion framework on these related learning objects. Similarly, two other types of heuristics are processed: suggestion and restriction heuristics. Suggestion heuristics offer relevant suggestion for the instructor to build a LOM attributes. Restriction heuristics specify constraints characterizing some LOM attributes. In a course authoring system based on graphs of LOM documents (e.g. [BPM03]), such a framework could be relevant to support the generation of LOM document. This system could also be extended to automatically generate queries to learning object repositories.

References

[BPM03] Nelson A. Baloian, José A. Pino, and Olivier Motelet. Collaborative authoring, use, and reuse of learning material in a computer-integrated classroom. In *CRIWG*, pages 199–207, 2003.

[1] Correspondence to: Olivier Motelet, DCC - Universidad de Chile, Avenida Blanco Encalada 2120, Tercer Piso, Santiago, Chile , C.P. 837-0459, Tel (+56 2) 678.4365, E-mail: omotelet@dcc.uchile.cl

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

From Representing the Knowledge to Offering Appropriate Remediation – a Road Map for Virtual Learning Process

Mehdi NAJJAR
Department of Computer Science, University of Sherbrooke,
2500, B$^{ld.}$ de l'Université, Sherbrooke QC J1k 2R1, Canada

Abstract. The paper describes a proposal of a '*from A to Z'* virtual learning process and its preliminary validation which consists of representing the knowledge, authoring graphically the subject-matter domain, modelling the students' believes and offering to each learner a personalised suitable feedback**.**

An important technological concept is being considered by an increasing number of universities and revolves about the idea of virtual learning. Nevertheless, several related key issues should be addressed, such as (1) the necessity to accurately represent the knowledge of the taught domain and the one handled and used by learners when interacting with the teaching material; and (2) the need to have tools which ease representing and modeling that knowledge and which are used by professors without the obligation of high capabilities in computer science at their disposal. These crucial points emerge the importance to exploit a representational model which offers structures that are closer to those recognised by psychology and cognitive science regarding the human learning processes. Especially, if one wishes to develop educational systems capable to adapt contents to the student profile and its needs and to provide tailored aid to learners according to their cognitive states.

To approach the mentioned issues, the broader aim of the research discussed in the paper is (1) to suggest a formal model of knowledge representation that is inspired from psychology cognitive theories, (2) to facilitate modelling the domain knowledge via user-centred graphical authoring tools which are "*life-complicated free*" and (3) to propose appropriate remediation and suitable suggestion mechanisms applied to help students engaged in learning activities through virtual learning environments. The paper is organised as follows. Section 1 describes the proposed theoretical model of knowledge representation and puts one's finger on some of its originalities. Section 2, presents an authoring tool prototype which offers the opportunity to model graphically the knowledge according to the proposed model. The graphic specification is transposed automatically into related XML files which are generated to serve as a knowledge support for a tutor reasoning purpose. Section 3 introduces a learning environment prototype (LEP) designed in order to exemplify educational systems which use teaching material specified via the authoring tool. The subject-matter domain of the LEP is the algebraic boolean expressions and their simplification by means of reduction rules, generally taught to undergraduate students. Preliminary experiments, made with students in computer science and in mathematics, are depicted in section 4. Finally, current developments are announced in section 5.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Authoring Ideas for Developing Structural Communication Exercises

Robinson V. Noronha*
Technological Institute of Aeronautics - ITA, SP, Brazil
rvida@cefetpr.br

Summary

The process of creating instructional activities based on ill-structured problems exercises (ISPs) is an enterprise that may expend a long time period and author's skills. Some of the authors has not the necessary skills to create this type of instructional activities such as some novice teachers. There is a high degree of uncertainty about this authoring process and it still remain a very costly and hard task. However there are some pedagogical techniques such as Structural Communication (SC) that could be used to structure and organize the domain knowledge and aid to produce this type of instructional activities. This paper discusses some ideas of computer tools and process to help the authoring activities of a SC unit. The purpose of these ideas is partially reduce some of the authoring difficulties.

Extractor of Keywords and Phrases (EKP) and Discussion Guide Generator (DGG) are two authoring ideas to aid the author. These ideas was implemented and will compose a suite of authoring tools. The EKP selects, extracts and sorts some sentences from a text source (Presentation section of a SC unit). These sentences are candidate elements to compose a grid, the Matrix Response Section of the SC unit. The student uses this grid to compose his/her solutions to an ill-structured problem.

The DGG creates some rules based on the result of these selected sentences. These rules are based on some defined meta-strategies, instructional goals and keywords of knowledge. The first meta-strategy concerns about how to foresee a possible student's solutions to a ISP. The second strategy tries to identify which concepts or keywords could be source of misunderstanding. The last strategy tries to identify student's misconceptions and gaps of his/her knowledge. These created rules are candidates to be used to conduct a deep analysis of students knowledge during the ill-structured problem solution process through SC unit. These rules will compose the Discussion Guide section of a SC unit.

Some of authoring difficulties to produce instructional activities with SC require resources or helpful ideas. These ideas could help inexpert authors during the authoring process. The author's skill to foresee the possible learner's solutions is not anymore essential during authoring process. DGG help author to “foresee” them. These ideas could constraints a creative author, or not? No, they cannot, because the creative author could refine the created rules to "foresee" the new student's solutions. The author exercises his/her abilities when he/she agrees or not agrees or changes the set of Matrix Response elements and set of Discussion Guide rules. Three meta-strategies also guide the author to create the Discussion Guide feedback message.

* On leave from CEFET/PR Brazil

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

An Orientation towards Social Interaction: Implications for Active Support

Abstract: Collaborative technologies mediate participants' communicative interactions. Because interaction data is stored, it can be used for further analysis. This implies that collaborative technologies can be extended with models that analyse interaction data and provide participants with 'on the spot' information about their performance. These models are based on some formal analysis that aggregates interaction data into meaningful information. However, we state that such a model should not solely focus on the interaction between learners, but that a broader understanding of learner-technology interaction is a prerequisite for development of 'active' support. Moreover, research findings indicate that interaction rules are not stable, but arise and evolve during interaction. The active system should be able to transform data to useful information, and moreover, it has to deal with changes in the way that rules and resources govern users' actions and interactions. This paper presents an exploratory orientation towards learner-technology interaction from a social-conceptual perspective, and discusses some implications for active support.

Keywords: Active support, Learner-technology interaction, and Structuration theory.

Maarten OVERDIJK and Wouter van DIGGELEN
Department of Educational Sciences, Utrecht University
Heidelberglaan 1, 3584 CS, Utrecht, The Netherlands
+31 30 253 3765
m.overdijk@fss.uu.nl / w.vandiggelen@fss.uu.nl

Structures and mediation: a critical account

Peoples' discourse is not driven by objects in the world, but by underlying structures like *e.g.* internal systems of meaning, modes of production, and inherent linguistic tendencies [1]. Structures are made available through cultural artefacts, and carry interfaces that mediate peoples' action. Broadly speaking, the term ′artefact′ comprises all culturally produced tools, such as *e.g.* sign-systems – also expressed in language, architecture, ICT's or embedded technology. What 'structure' is, and how we should conceptualize it, is subject to different interpretations. Structure is traditionally seen as a stable and somewhat rigid construct that determines human action in a constraining way. Recently, scholars have emphasized the dynamic, discursive construction of structure. In this view, agents draw from structures in order to engage in interaction. Structure is hereby seen as constraining *and* enabling. Interacting agents are capable of reproducing and also of producing structures. This discursive relation between agency and structure accounts for *changes* in a social system.

Structuration theory (ST) attempts to explain how action in groups becomes structured through interaction [2]. It recognizes a *social system* of interacting human agents and *structure,* in order to explain how practices develop and persist over time and space. Structuration is the process by which systems are produced and reproduced through members' use of rules and resources. Structure is seen as both the medium and outcome of the conduct it – recursively, organizes. In ST, s*tructure is not a stable entity*. Social systems do not have structures but rather exhibit structural properties. Structuring properties are both enabling and constraining, and allow the binding of time-space in the system; these properties "make it possible for discernibly similar social practices to exist across varying spans of time and space and lend them systemic form" [2]. Subjects are seen as "knowledgeable agents" that reflexively interact with the rules and resources that are made available through structuring properties in the environment. This upgrade of agency in ST has drawn attention to the ways in which agents actively shape their environment. Although the theory focuses on the relation between agency and structure on the macro-level, it offers some illuminating notions to account for change and stability in the relation between agency and structure on the micro-level. The focus is then on the practice of the small-group as a part of a larger social system.

Collaborative technologies provide opportunities for specific types of communicative action because they make certain rules and resources available to the actors as they carry out their discourse. The technology contains certain structuring properties that concern *e.g.* navigation, organisation of participation in activity, possible (communicative) actions in the system, or modality of expression and representation. Collaborative technologies hereby shape the interaction: they enable the occurrence of certain actions and constrain others. However, in many cases it are not so much the structural properties of a medium that determine the nature of its use in practice, but rather the rules and conventions that result from ongoing discursive application. Moreover, research findings [3] indicate that interaction rules are not stable, but arise and evolve during interaction. The structuring qualities of rules can be studied in respect of the forming, sustaining, termination and reforming of discourse [2].

The theoretical exploration presented in this paper indicates that appropriation of rules and resources that govern users' actions and interactions is a *dynamic* process. An interaction model for active support should be able to deal with *changes* in the appropriation of rules and resources. In other words, it should not assume stable structure. It should be able to deal with stability from a concept that foregrounds change. Application of ideas from Structuration theory to the study of interaction offers a tentative approach to such a dynamic view, and has implications for interaction analysis.

References

[1] Gergen, K. (1994). *Realities and Relationships: Soundings in Social Construction.* Cambridge, Massachusetts, Harvard University Press.

[2] Giddens, A. (1984). *The Constitution of Society: Outline of the Theory of Structuration.* Polity Press, Cambridge.

[3] Diggelen, van, W., Overdijk, M. and Andriessen, J. (2004), Constructing an argumentative map together: Organizing principles and their application. *Paper presented at the Dutch Educational Research Conference (ORD), Utrecht,* http://edu.fss.uu.nl/ord/homepage.htm.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Designing Culturally Authentic Pedagogical Agents

Yolanda Rankin
Northwestern University, Evanston, Illinois, USA

Ethnicity has been defined primarily by the physical appearance of pedagogical agents. Research has shown that shared ethnicity between the agent and the user reflects in a positive perception of the agent's capabilities and provides motivation for learning tasks for students of color (Baylor, 2005; Nass et al., 2000). However, research has failed to examine what kinds of implementations of pedagogical agents are the most authentic representations. As designers of educational technology, pedagogical agents should be designed to reflect authentic portrayals of ethnic groups. I argue that ethnicity includes more than physical appearance but encompasses verbal and nonverbal behaviors as well. I observe verbal and nonverbal behaviors of African American preschool children as they participate in storytelling. I have adapted Cassell's (2000) methodology to the construction of ethnically authentic pedagogical agents. My research goal is to design culturally authentic agents that bridge the gap between language skills practiced outside the classroom setting and those language skills required in the classroom.

I examined the behavior of seven African American children between the ages of 5 and 7 years old telling stories while playing with toys in a wooden castle. The children told stories for fifteen to twenty minutes; the storytelling sessions were videotaped and transcribed. To analyze the content of speech, the collected stories were evaluated for presence of AAVE discourse features (Green, 2001). I found more than twelve verbal features of AAVE and grouped them into four categories: phonology (e.g. deletion of word-final single consonant after a vowel), syntax (e.g. absence of copula for present tense), lexicon (e.g. use of *finna* to mark the immediate future) and narrative style (e.g. raised pitch for impersonation of characters). In addition, I observed and identified ethnic nonverbal communication including gestures, rolling eyes, rocking head and neck movement and body position. Based upon the observations, I selected a subset of AAVE discourse features to be implemented in the pedagogical agent named Alex. The implementation follows our previous Flash implementations of the virtual peer.

In conclusion, I have attempted to create a prototype of an authentic virtual representation of an African American child. Future research efforts raise the following question: Will children perceive Alex as a being African American based upon verbal and nonverbal behaviors? Further research is planned to evaluate the affects of culturally authentic pedagogical agents on African American children's early language acquisition skills. This brings us one step closer to designing and implementing culturally authentic pedagogical agents.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Incorporation of Learning Objects and Learning Style - Metadata Support for Adaptive Pedagogical Agent Systems

Shanghua Sun
Department of Computer Science
University of Warwick, Coventry CV4 7AL, United Kingdom

Learning objects have been increasingly used in pedagogical systems, but effective pedagogic strategies to support adaptive learning are still lacking. There are many metadata strategies for learning object design and categorization, but research about incorporating real learning objects with learning style schemes into education systems is rare.

We have developed a pedagogical agent-based system, in which agent technology provides a dynamic adaptation not only of domain knowledge but also of the behaviour of individual learners. The system is student-centred, adaptive and dynamic, and our approach takes a multi-disciplinary approach, combining learning theory with agent-based systems. In contrast to other agent-based pedagogic architectures, the incorporation of learning objects and learning style schemes form the pedagogic foundation for adaptivity.

The learning style theory we have adopted in the system is the Felder-Silverman Learning Style Model [1]. In addition to the descriptions in the existing metadata standards, the learning object metadata incorporates a *dimension description*, suggesting for each of the four learning style dimensions the extent of each object's suitability on a five-point scale. The system stores each student's current learning style (which may change over time), and in the metadata the style dimension description of each learning object as co-ordinates in the four-dimensional space. The algorithm used to deliver learning objects to students involves matching the style attributes of (appropriate) learning objects to the current style preferences of the individual student. The system then searches the repository of learning objects to fetch appropriate learning objects with similar (but not necessarily identical) dimensional descriptions. These are supported by agent technology to realize the algorithm and implement the process. The objects are then presented to the student, and subsequent interactions between the student and the learning objects may be used to modify the student's learning style attributes.

The evaluation suggests that the approach is appropriate for the pedagogical agent system. Current and future work includes further investigation of the granularity of the learning object category, and optimising the system architecture to enhance its effectiveness and efficiency. For more information, we refer readers to the paper in the proceeding of Young Researcher Track.

References

[1] R. M. Felder and J.E. Spurlin, Applications, Reliability, and Validity of the Index of Learning Styles, *Int. Journal of Engineering Education*, 21(1), (2005), 103-112.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Enhancing Collaborative Learning through the use of a Group Model based on the Zone of Proximal Development

Nilubon TONGCHAI[1]
The SCRE Centre, Faculty of Education,
University of Glasgow, 11 Eldon Street, Glasgow, G3 6NH

Collaborative Learning is seen as a good way to encourage peers to learn and to teach each other whereas Open Learner Modelling can help learners to improve their performance and their understanding using high-level indicators to monitor, and represent, the state of their learning. This research seeks to apply both concepts of Collaborative Learning and Open Learner Modelling. Less has been done with Group Open Learner Models (GOLMs) though the idea has potential [1].

The group model will borrow ideas from both Paiva's work [2] and PairSM [3] to generate the group model taking the ZPD concept into account. In this work we would like to know whether the inspection of the GOLM improves learning. To answer that question, the value from the difference between the Ideal GLM[2] and GLM[3] is compared. If learners see a pie-chart and perform better than learners who cannot see the group model, we may be able to conclude that a group model is effective for collaborative learning.

A prototype will be built to demonstrate the working of the model and it is expected to use fuzzy logic for dealing with the uncertainty in such a model. After the model has been developed further, the approach above will be implemented, tested and revised prior to developing the model used for the final study with learners.

The work in this thesis aims to encourage students to obtain an advantage from both collaborative learning and the use of an Open Learner Model to try to prove that the result of collaborative learning with a group model capitalise Open Learner Model allows the learner to get a higher score than when unable to inspect the group model. Now we are in the process of simplifying the group model taking the ZPD into account and using fuzzy logic as a technique to generate values representing group knowledge After the hypothesis described above is tested, the next question for this work is ‘In what ways is a Group Learner Model better than an Individual Learner Model?’

References

[1] Zapata-Rivera, J.-D. and J.E. Greer. (2004). Interacting with Inspectable Bayesian Student Models. *International Journal of Artificial Intelligence in Education,* 14, 1-37.

[2] Paiva, A. (1997). Learner Modelling for Collaborative Learning Environment. In B. du Boulay, & R.Mizoguchi (Eds.) *Artificial Intelligence in Education* (pp. 215-222). Amsterdam: IOS Press.

[3] Bull, S. and Smith, M.(1997). A Pair of Student Models to Encourage Collaboration. In A. Jameson, C. Paris & C. Tasso (Eds.) *Proceedings of theSixth International Conference on User Modeling*. Springer.

[1] The PhD student is under the supervision of Dr.Paul Brna.

[2] Ideal GLM is a model of the group that is generated from the performance of individual learners

[3] the group model that reflect when learner1 and learner2 collaboratively perform to solve the group task.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Tutorial Planning: Adapting Course Generation to Today's Needs

Carsten ULLRICH
German Research Center for Artificial Intelligence (DFKI GmbH),
Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany

Most of today's course generation (the automatic assembly of sequences of learning objects, e.g., [2]) does not allow the declarative representation of pedagogical knowledge as possible with, e.g., [3]. In my work, I develop a framework that adapts these approaches to today's needs. Within this framework several of today's problems are solved, such as the integration of distributed content and e-learning services within a course, dynamic adaptivity of a generated course, new forms of interaction, and offering course generation as a service.

A declarative and generic representation of pedagogical knowledge as advocated in [3] distinguishes between the tutorial tasks to be achieved (e.g., `teachConcept` or `provide AdequateExercise`) and different methods of achieving these tasks (e.g., using a problem-based educational strategy, and depending on the learner's characteristics). In my approach the pedagogical knowledge is executed by a hierarchical task network planner [1]. The result of the planning is a sequence of learning objects (a content structure). As tasks represent a vast range of pedagogical goals, the size of the sequence ranges from a single element to a complete curriculum.

The problem of how to provide dynamic adaptivity of a generated course can serve to exemplify my approach. Course generation faces the dilemma that early course generation cannot take into account how capabilities of the learner actually change. Still, generating a course as early as possible supports orientation and self-organisation of the learning process. A different solution as plan repair is lazy task execution. In this approach, planning may stop at the level of specially marked tasks (lazy tasks). These tasks are inserted in the content structure just like any other learning objects. When the learner first visits a page that contains a lazy task, the task in the content structure is passed to the tutorial planner. The resulting learning objects replace the task in the course structure for good (hence, when the page is revisited, the elements do not change, which avoids confusion of the learner). This means a course is partly static, partly dynamic. Lazy task execution offers new possibilities for authors, too. An author can define a course structure, where parts of her course are predefined, and others dynamically computed taking the learner model into account. In this way, an author can profit from the best of both worlds: she can compose parts of the course by hand and at the same time profit from the adaptive features of the tutorial planner.

References

[1] K. Erol, J. Hendler, and D. S. Nau. HTN planning: Complexity and expressivity. In *Proc. of the 12th National Conference on Artificial Intelligence (AAAI-94)*, volume 2, pages 1123–1128, Seattle, Washington, USA, 1994. AAAI Press/MIT Press.

[2] P. Karampiperis and D. Sampson. Adaptive instructional planning using ontologies. In *Proc. of the 4th IEEE International Conference on Advanced LearningTechnologies*, pages 126–130, 2004.

[3] K. van Marcke. Instructional expertise. In C. Frasson, G. Gauthier, and G. McCalla, editors, *Proc. of the 2cd International Conference on Intelligent Tutoring Systems*, number 608 in Lecture Notes in Computer Science, pages 234–243. Springer, 1992.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Mutual Peer Tutoring: A Collaborative Addition to the Cognitive Tutor Algebra-1

Erin WALKER
Carnegie Mellon University, Pittsburgh, PA, USA
erinwalk@andrew.cmu.edu

Use of the Cognitive Tutor Algebra-1 (CT) improves learning by about one standard deviation over traditional classroom instruction. Students might show further learning gains if their use of cognitive tutors was combined with collaborative activities. Because collaboration is most effective when certain positive behaviors are encouraged, we plan to structure student interaction, using a peer tutoring script (PTS). We propose to integrate the script into the existing CT framework and develop a collaborative tutor for the PTS.

In the PTS approach, students are placed in homogeneous dyads with respect to ability and take turns being the peer tutor (the person doing the tutoring) and the peer tutee (the person being tutored). The script has three phases: a preparation phase, a collaboration phase, and a meta-evaluation phase. In the preparation phase, peer tutors prepare to teach the peer tutee in two ways: 1) They solve an algebra problem with the aid of the CT, and 2) they match each problem step to the cognitive skill required to complete it. In the collaboration phase, peer tutors help the peer tutee solve the same algebra problem by correcting student answers, rating the tutee's skill mastery, and providing feedback. The CT will also be used in this process to monitor and tutor the collaboration between the students. During the meta-evaluation phase, the students have a structured discussion of the skills that they used. Students should show learning gains after using this script because of their engagement in positive collaborative behaviors and in monitoring their own and others' skills.

To implement the PTS I am modifying the interface and cognitive model of the CT. This approach will reduce the cognitive burden on the students, as they will already be familiar with the interface and understand how tutoring works within its context. There are also technical advantages to this approach because existing CT components can be repurposed to aid in the implementation of the cognitive model for collaboration. The addition of the PTS to the CT will involve design and implementation phases, followed by an experimental phase to evaluate the suitability of the script for deployment.

In an early step toward developing the collaborative tutor for the PTS, we conducted a pen-and-paper pretest with two dyads of middle-school students. Observation revealed that the interaction process was relevant. For example, a model needs to be developed for the three stages of providing explanations: recognizing the need for help, giving hints, and confirming understanding. Surprisingly, tutoring content was troublesome for the students and also needs to be supported by the collaborative tutor. Peer tutors would often struggle with generating explanations that could be understood by the tutee. We intend to use further pretest data to gather information about positive student actions and common student errors that could then form the basis for the collaborative model.

Acknowledgements
This research is funded by the National Science Foundation, Award #0354420. I thank Ken Koedinger, Bruce McLaren, Nikol Rummel, and Mindy Kalchman for their help.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Enhancing Learning through a Model of Affect

Amali WEERASINGHE
Intelligent Computer Tutoring Group
Department of Computer Science, University of Canterbury
Private Bag 4800, Christchurch, New Zealand
acw51@student.canterbury.ac.nz

The effectiveness of human one-to-one tutoring is largely due to the tutor's ability to adapt the tutorial strategy to the students' emotional and cognitive states. Even though tutoring systems were developed with the aim of providing the experience of human one-to-one tutoring to masses of students in an economical way, using learners' emotional states to adapt tutorial strategies have been ignored until very recently. As a result, researchers still focus on generating affective models and evaluating them. To the best of our knowledge, a model of affect is yet to be used to improve the objective performance of learners. This paper proposes an initial study to understand how human tutors adapt their teaching strategies based on the affective needs of students. The findings of the study will be used to investigate how these strategies could be incorporated into an existing tutoring system which can then adapt to the learner's affect and cognitive models.

Several researchers have pointed out that it is more important to focus on using the student model to enhance the effectiveness of the pedagogical process, than building a highly accurate student model that models everything about the student. Therefore, we are interested in investigating how a model of affect can be used to improve learning. We choose to focus on using the affective model to develop an effective problem selection strategy because most ITSs employ adaptive problem selection based only on the cognitive model, which may result in problems being too easy or too hard for students. This may occur due to factors like how much guessing was involved in generating the solution, how confident she was about the solution, how motivated she was etc., which are not captured in the student's cognitive model. Therefore, using both cognitive and affective models can potentially increase the effectiveness of a problem selection strategy, which in turn can improve the learners' motivation to interact with the system.

As we want to explore how emotional states could be used to adapt the tutoring strategies, we propose to conduct a study to understand how human tutors respond to learners' affective states. The objectives of the study are to understand how human tutors identify the emotional states of students during learning and how they adapt tutoring strategies in each situation. Participants will be students enrolled in an introductory database course at the University of Canterbury. As we want to explore general tutoring strategies, we plan to use four existing tutoring systems developed by our research group. Several tutors will observe students' interactions. All sessions will be videotaped. Based on the study, we want to explore how this adaptation of tutorial strategies can be incorporated into an intelligent tutoring system.

© 2005 The authors. All rights reserved.

Understanding the Locus of Modality Effects and How to Effectively Design Multimedia Instructional Materials

Jesse S. Zolna
Department of Psychology, Georgia Institute of Technology

Abstract

AIED learning systems sometimes employ multimedia instructional materials that leverage technology to replace instructional text with narrations. This can provide cognitive advantages and disadvantages to learners. The goal of this study is to improve principals of information design that cater to human information processing. Prior research in educational psychology has focused on facilitating learning by presenting information in two modalities (auditory and visual) to increase perceptual information flow. It is hypothesized that similar effects might also occur during cognitive manipulations (e.g., extended storage and fact association). The described study separates perceptual information effects from those of cognitive operations by presenting auditory and visual information separately. The typical multimedia effect was not found, but other influences on learning were observed. An understanding of these other causes will help us create a more complete picture of what producers of multimedia learning materials should consider during design.

Summary

Contemporary technology is increasingly employed to improve the efficiency of educational instruction. Educational psychologists have been trying to understand how *multimedia instructional materials*, that is presenting to-be-learned information in more than one modality, can improve learning [1;2]. The goal of this study is to advance the limited knowledge associated with mixing media ingredients that best cater to the strengths and limitations of human information processing. Research related to instructional design has proposed that controlling the processing demand needed in multimedia learning environments might be achieved by spreading information among working memory stores [1;2]. The focus of these explanations have been on perceptual level encoding (i.e., transition from the sensory store), creating information design recommendations that center on the presentation of multimodal information. They have deemphasized how the two streams of information influence the active processing of new information. The two influences, that is on perceptual encoding and active processing, may be separable, each influential for learning. If so, designing multimedia interfaces with considerations for only perceptual effects, as has been common in the past, may be incomplete.

Non-verbal (or visual-spatial) and verbal (or auditory) internal representations often correspond to diagrammatic and descriptive external representations, respectively. However, visually and auditorily presented information included in multimedia learning environments correspond imperfectly to this division of internal representations. Research investigating multimedia instructional materials in light of psychological models [3;4;5] will define internal representations by more than just materials' external representations. In an experiment, typical multimedia learning effects were not found. The next steps are to understand human information processing based on the effects of modality for both internal and external representations of information, and consequently to make suggestions to designers of multimedia information.

References

[1] Mayer, R. (2001) *Multimedia Learning.* Boston: Cambridge University Press.

[2] Sweller, J. (1999). *Instructional Design.* Melbourne: ACER Press.

[3] Baddeley, A., & Hitch, G.J. (1994). Developments in the concept of Working Memory. *Neurosychology*, *8*(4), 485-493.

[4] Paivio, A. (1986). *Mental representations: A dual coding approach.* New York: Oxford University Press

[5] Wickens, C. D. (2002). Multiple resources and performance prediction. Theoretical Issues in Ergonomic Science, 3(2), 159-177.

This page intentionally left blank

Panels

This page intentionally left blank

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Pedagogical agent research and development: Next steps and future possibilities

Amy L. BAYLOR
Director, Center for Research of Innovative Technologies for Learning (RITL)
Florida State University
http://ritl.fsu.edu
baylor@coe.fsu.edu

Ron COLE
Director, Center for Spoken Language Research (CSLR)
Univ. of Colorado at Boulder
cole@cslr.colorado.edu

Arthur GRAESSER
Co-Director, Institute for Intelligent Systems (IIS)
University of Memphis
a-graesser@memphis.edu

W. Lewis JOHNSON
Director, Center for Advanced Research in Technology for Education (CARTE)
University of Southern California
johnson@isi.edu

Abstract. The purpose of this interdisciplinary panel of leading pedagogical agent researchers is to discuss issues regarding implementation of agents as "simulated humans," pedagogical agent affordances/constraints, and future research and development possibilities.

Introduction

Pedagogical agent research and development has made significant strides over the past few years, incorporating animated computer characters that are increasingly more realistic and human-like with respect to their dialogue, appearance, animation and the instructional outcomes they produce. Given the rapid growth and convergence of knowledge and technologies in areas of cognitive science (how people learn, how effective teachers teach), computing / networking and human communication technologies, the vision of accessible and affordable intelligent tutoring systems that use virtual teachers to help students achieve deep and useful knowledge has moved from fantasy to emerging reality. This panel will build on other recent discussions (including an NSF – supported "Virtual Humans Workshop") to assess the current state of knowledge of pedagogical agents, and discuss the science and technologies required to accelerate progress in this field.

1. Organization of Panel

A brief overview of the construct of "pedagogical agent" will be presented together with a review of pedagogical agent effectiveness for different learning outcomes (e.g., content acquisition, metacognition, motivation). The panel discussion will focus on four key sets of questions (listed below), for which each panellist will present a brief prepared response. Following each of the four panellists' responses, there will be time for broader discussion of the question among the panellists.

1. Definitions:
   - What constitutes a pedagogical agent (e.g., message, voice, image, animation, intelligence, interactivity)?
   - Is the agent interface enough to constitute a pedagogical agent?
   - How intelligent (cognitively, affectively, and/or socially) should pedagogical agents be?
2. Human-likeness:
   - How human-like should agents be with respect to the different modalities? What new technologies and knowledge (e.g. social dynamics of face to face tutoring) are required to make pedagogical agents look and act like human teachers?
   - How can we best exploit the *human-like* benefits (e,g., affective responses) of pedagogical agents together with their benefits as a *technology* (e.g., control, adaptivity)
3. Instructional affordances (and constraints):
   - What new possibilities can pedagogical agents provide? (e.g., unique instructional strategies, providing a social presence when online instructor is absent, employing multiple agents to represent different perspectives)
   - What constraints exist? (e.g., user expectations and stereotypes)
4. The future:
   - What are the main technological challenges and research breakthroughs required to invent virtual humans, and when can we expect these challenges to be met?
   - What multidisciplinary research is required to invent pedagogical agents that behave like sensitive and effective human teachers? When might we expect a virtual teacher to pass a Turing test, e.g., teach a student to read or solve a physics problem as if it were an expert human tutor? What would this test look like?
   - What are some new possibilities for agents (e.g., in different artefacts and settings, in different roles/functions, to serve as simulated instructors and test-beds for controlled research)?

This page intentionally left blank

Tutorials

This page intentionally left blank

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Evaluation methods for learning environments

Shaaron Ainsworth
School of Psychology and Learning Sciences Research Institute,
University of Nottingham, Nottingham, UK

This tutorial explores the issue of evaluation in AIED. The importance of evaluating AIED systems is increasingly recognised. Yet, there is no single right way to evaluate a complex learning environment. This tutorial will emphasize how to develop a practical toolkit of evaluation methodologies by examining classic case studies of evaluations, show how techniques from other areas can be applied in AIED and examine common mistakes. Key issues include:

- the goals of evaluation (e.g. usability, learning outcomes, learning efficiency, informing theory),
- choosing methods for data capture and analysis,
- appropriate designs,
- what is an appropriate form of comparison?
- and the costs and benefits of evaluating "in the wild."

Audience: This is an introductory tutorial intended for researchers with a variety of backgrounds.

Presentation: Slides interspersed with demonstrations and discussions. Working in groups participants will design their own evaluation plans for a system during the course of the session.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Rapid development of computer-based tutors with the Cognitive Tutor Authoring Tools (CTAT)

Vincent Aleven, Bruce McLaren and Ken Koedinger
Carnegie Mellon University
Pittsburgh, Pennsylvania USA

The use of authoring tools to make the development of intelligent tutors easier and more efficient is an on-going and important topic within the AI & Ed community. This tutorial provides hands-on experience with one particular tool suite, the Cognitive Tutor Authoring Tools (CTAT). These tools support the development and delivery (including web delivery) of two types of tutors: problem-specific Pseudo Tutors, which are very easy to build, and Cognitive Tutors, which are harder to build but more general, having a cognitive model of a competent student's skills. Cognitive Tutors have a long and successful track record: they are currently in use in over 2000 US high schools. The CTAT tools are based on techniques of programming by demonstration and machine learning. The tutorial will provide a combination of lectures, demonstrations, and a good amount of hands-on work with the CTAT tool suite. CTAT is available for free for research and educational purposes (see http://ctat.pact.cs.cmu.edu).

The target audience includes

- ITS Researchers and developers looking for better authoring tools
- Educators (e.g. college level professors) with some technical background interested in developing on-line exercises for their courses
- Researchers in education or educational technology interested in using tutoring systems as a research platform to explore hypotheses about learning and/or instruction.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Some New Perspectives on Learning Companion Research

Tak-Wai Chan
National Central University, Taiwan

Learning companions, a concept proposed in 1988, were originally intended to be an alternative model of intelligent tutoring systems. This concept has recently drawn a rapid growth of interest while the research has been going along with generation of a variety of names such as virtual character, virtual peer, pedagogical agent, trouble maker, teachable agent, animal companion, and so forth. A number of research and technological advancements, including affective learning, social learning, human media interaction, new views on student modeling, increase of storage capacity, Internet, wireless and mobile technologies, ubiquitous computing, digital tangibles, and so forth, are driving learning companion research to a new plateau. This tutorial intends to give an account of these new perspectives and to shed light on a possible research agenda on the ultimate goal of learning companion research — building a lifelong learning companion.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Education and the Semantic Web

Vladan Devedžić
University of Belgrade, Serbia and Montenegro

The goals of this tutorial are to present important theoretical and practical advances of the Semantic Web technology and to show its effects on education and educational applications. More specifically, important objectives of the tutorial are to explain the benefits the Semantic Web brings to Web-based education, and to survey current efforts in the AIED community related to applying Semantic Web technology in education. Some of the topics to be covered during the tutorial include: ontologies, Semantic Web languages, services and tools, educational servers, architectural aspects of the Semantic Web AIED applications, learner modeling and The Semantic Web, instructional design and The Semantic Web, and semantic annotation of learning objects.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Building Intelligent Learning Environments: Bridging Research and Practice

Beverly Park Woolf
University of Massachusetts, Amherst, Massachusetts, USA

This tutorial will bring together theory and practice about technology and learning science and take the next step toward developing intelligent learning environments. We will discuss dozens of example tutors and present a wealth of tools and methodologies, many taken from mathematics and science education, to help participants design and build their own intelligent learning environments. Discussions will focus on linking theory in learning systems, artificial intelligence, cognitive science and education with practice in writing specifications for an intelligent tutor.

Participants are encouraged to select an academic domain in which they want to build an intelligent learning environment and the group will break into teams several times during the tutorial to solve design and specification problems. The tutorial will provide a suite of tools and a toolkit for general work productivity and will emphasize a team-oriented, project based approach. We will share tutor techniques and identify some invariant principles behind successful approaches, while formalizing design knowledge within a class of exemplary environments in reusable form.

This page intentionally left blank

Workshops

This page intentionally left blank

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Student Modeling for Language Tutors

Sherman ALPERT[1] and Joseph E. BECK[2]
[1] IBM T.J. Watson Research Center
[2] Center for Automated Learning and Discovery, Carnegie Mellon University
salpert@us.ibm.com, joseph.beck@cmu.edu

Abstract. Student modeling is of great importance in intelligent tutoring and intelligent educational assessment applications. However, student modeling for computer-assisted language learning (CALL) applications differs from classic student modeling in several key ways, including the lack of observable intermediate steps (behavioral or cognitive) involved in successful performance. This workshop will focus on student modeling for intelligent CALL applications, addressing such domains as reading decoding and reading and spoken language comprehension. Domains of interest include both primary (L1) and second language (L2) learning. Hence, the workshop will address questions related to student modeling for CALL, including what types of knowledge ought such a model contain, with what design rationale, and how might information about the user's knowledge be obtained and/or inferred in a CALL context?

Topics and goals

Student modeling is of great importance in intelligent tutoring and intelligent educational diagnostic and assessment applications. Modeling and dynamically tracking a student's knowledge state are fundamental to the performance of such applications. However, student modeling in CALL applications differs from more "classic" student modeling in other domains in three key ways:

1. It is difficult to determine the reasons for successes and errors in student responses. In classic ITS domains (e.g., math and physics), the interaction with the tutor may require students to demonstrate intermediate steps. For performance in language domains, much more learner behavior and knowledge is hidden, and having learners demonstrate intermediate steps is difficult or perhaps impossible, and at any rate may not be natural behavior. (How) Can a language tutor reason about the cause of a student mistake? (How) Can a language tutor make attributions regarding a student's knowledge state based on overt behavior?
2. Cognitive modeling is harder in language tutors. A standard approach for building a cognitive task model is to use think-aloud protocols. Asking novices to verbalize their problem solving processes while trying to read and comprehend text is not a fruitful endeavor. How then can we construct problem solving models? Can existing psychological models of reading be adapted and used by computer tutors?
3. It may be difficult to accurately score student responses. For example, in tutors that use automated speech recognition (ASR), whether the student's response is correct cannot be determined with certainty. In contrast, in classic tutoring systems scoring the student's response is relatively easy. How can scoring inaccuracies be overcome to reason about the students' proficiencies?

This workshop discusses attempts at solutions to these and related problems in student modeling for language tutors.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

International Workshop on Applications of Semantic Web Technologies for E-Learning (SW-EL'05)

Lora AROYO[1] and Darina DICHEVA[2]
[1]*Department of Computing Science, Eindhoven University of Technology*
PO Box 513, 5600 MD Eindhoven, The Netherlands
l.m.aroyo@tue.nl
[2]*Department of Computer Science, Winston-Salem State University*
601 Martin Luther King, Jr. Drive, Winston Salem, N.C. 27110, USA
dichevad@wssu.edu

Abstract. The SW-EL'05 workshop at AIED'05 covers topics related to the use of ontologies for knowledge representation in intelligent educational systems, modularised and standardized architectures, achievement of interoperability between intelligent learning applications, sharable user models and knowledge components and support for authoring of intelligent educational systems. Two focus-sessions are included in the workshop:

1) Application of Semantic Web technologies for Adaptive Learning Systems, which focuses on personalization and adaptation in educational systems (flexible user models), ontology-based reasoning for personalising the educational Semantic Web, and on techniques and methods to capture and employ learner semantics.

2) Application of Semantic Web technologies for Educational Information Systems, which focuses on Semantic Web-based indexing/annotation of educational content (incl. individual and community based), on ontology-based information browsing and retrieval and Semantic Web/ontology based recommender systems.

Papers presented in the workshop illustrate Semantic Web-based methods, techniques, and tools for building and sharing educational content, models of users, and personalisation components; services in the context of intelligent educational systems (i.e. authoring service, user modelling service, etc.) and ontology evolution, versioning and consistency. A key part of the reported results are related to empirical research on Intelligent Educational Systems presenting real-world systems and case studies and providing community and individual support by using Semantic Web-technologies and ontologies. The workshop is also a forum for presenting research performed within the context of the KALEIDOSCOPE and PROLEARN network of excellences.

Other editions of the SW-EL workshop include:

- SW-EL'05 at ICALT'05, Kaohsiung, Taiwan
- SW-EL'05 at AIED'05, Amsterdam, The Netherlands
- SW-EL'05 at K-CAP'05, Banff, Canada
- SW-EL'04 at AH'04, Eindhoven, The Netherlands
- SW-EL'04 at ITS'04, Maceio, Brazil
- SW-EL'04 at ISWC'04, Hiroshima, Japan

General workshop web site: http://www.win.tue.nl/SW-EL/index.html

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Adaptive Systems for Web-Based Education: Tools and reusability

Peter Brusilovsky; University of Pittsburgh; peterb@mail.sis.pitt.edu
Ricardo Conejo; University of Málaga; conejo@lcc.uma.es
Eva Millán; University of Málaga; eva@lcc.uma.es

Motivation

Web-based education is currently a hot research and development area. Benefits of Web-based education are clear at hand: learners from all over the world can enroll in learning activities, communicate with other students or teachers, can discuss and control their learning progress - solely based on an internet-capable computer. A challenging research goal is to tailor the access to web-based education systems to the individual learners' needs, as determined by such factors as their previous knowledge on the subject, their learning style, their general attitude and/or their cultural or linguistic background. A number of Web-based adaptive and intelligent systems have been developed over the last 5 years. However, a larger variety of innovative systems can still be created and evaluated to provide a real difference in E-Learning.

The goal of this workshop is to provide a forum for the discussion of recent trends and perspectives in adaptive systems for web-based education, and thus to continue the series of workshops on this topic held at past conferences.

Topics

The list of topics includes, but is not limited to:

- Adaptive and intelligent web-based collaborative learning systems
- Web-based adaptive educational hypermedia
- Web-based Intelligent tutoring systems
- Adaptive Web-based testing
- Web-based Intelligent class monitoring systems
- Adaptive and intelligent information retrieval systems for web-based educational materials
- Personalization in educational digital libraries
- Architectures for adaptive web-based educational systems.
- Using machine learning techniques to improve the the outcomes of Web-based educational processes
- Using semantic web technologies for adaptive e-learning
- Reusability and self-organisation techniques for educational material
- Interoperability between tools and systems for adaptive e-learning
- Pedagogical approaches in web-based educational systems

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Usage analysis in learning systems

AIED2005 Workshop
(**http://lium-dpuls.iut-laval.univ-lemans.fr/aied-ws/**)

The topic of analyzing learning activities has attracted a lot of attention in recent years. In particular a number of techniques have been proposed by the AIED Community to collect and analyze data in technology supported learning activities. Understanding and taking into account usage of learning systems is now a growing topic of AIED Community, as recent events (ITS2004 workshop) and projects ("Design Patterns for Recording and Analyzing Usage in Learning Systems" work package of the European Kaleidoscope Network) have shown.

Learning systems need to track student usage and to analyze their activity in order to adapt dynamically the teaching strategy during a session and/or to modify contents, resources and scenario after the session to prepare the next one. These large amounts of student data can also offer material for further analysis using statistical, data mining or other techniques. The aims of this workshop are (1) to facilitate the sharing of approaches, problems and solutions adopted for usage analysis of learning systems and (2) to create a forum for collaboration and to develop an international community around this field of study.

The workshop will consist in presentations of refereed papers and posters, discussions and end with a forum led by a panel (Nicolas Balacheff, Ulrich Hoppe and Judy Kay) aimed at synthesizing workshop contributions and at identifying promising directions for future work.

Program Committee

Christophe Choquet, LIUM, University of Maine, France (co-chair)
Vanda Luengo, IMAG, University of Grenoble, France (co-chair)
Kalina Yacef, SIT, University of Sydney, Australia (co-chair)
Nicolas Balacheff, IMAG, University of Grenoble, France
Joseph Beck, Carnegie Mellon University, USA
Peter Brusilovsky, School of Information Sciences, University of Pittsburgh, USA
Elisabeth Delozanne, CRIP5, University of Paris 5, France
Angelique Dimitrakopoulou, Aegean University, Greece
Ulrich Hoppe, COLLIDE, University Duisburg Essen, Germany
Judy Kay, SIT, University of Sydney, Australia
Jean-Marc Labat, AIDA, Paris 6 University, France
Frank Linton, The Mitre Corporation, MA, USA
Agathe Merceron, Leonard de Vinci University, Paris, France
Tanja Mitrovic, University of Canterbury, Christchurch, New Zealand
Jack Mostow, School of Computer Science, Carnegie Mellon University, USA
Ana Paiva, INESC, Lisboa, Portugal.
Richard Thomas, University of Western Australia, WA, Australia
Pierre Tchounikine, LIUM, University of Maine, France
Felisa Verdejo, UNED, Madrid, Spain

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Workshop on Educational Games as Intelligent Learning Environments

Cristina Conati
Department of Computer Science, University of British Columbia,
2366 Main Mall, Vancouver, BC, V6T1Z4, Canada
{manske, conati}@cs.ubc.ca

Sowmya Ramachandran
Stottler Henke Associates, Inc,
951 Mariner's Island Blvd., Ste 360, San Mateo, CA 94404
Sowmya@stottlerhenke.com

Over the past decade there has been an increasing interest in electronic games as educational tools. Educational games are known to be very motivating and they can naturally embody important learning design principles like exploration, immersion, feedback, increasingly difficult challenges to master. However, there are mixed results on the actual pedagogical effectiveness of educational games, indicating that this effectiveness strongly depends upon preexisting students' traits such as meta cognitive skills and learning attitudes. These results are consistent with the mixed results on the effectiveness of exploratory learning environments, not surprisingly since most educational games are exploratory learning environments with a stronger focus of entertainment.

Artificial Intelligence is already playing a increasingly integral part in both non-educational game design, and the design of more effective exploratory learning environments. This workshop aims to explore if and how AI techniques can also help improve the scope and value of educational games.

The overall goal of the workshop is to bring together people who are interested in exploring how to integrate games with intelligent educational technology, to review the state-of-the –art, and formulate directions for further exploration.

Some of the questions that the workshop aims to address include: (1) are some genres of games more effective at producing learning outcomes? (2) How do learners ' individual differences (cognitive, meta-cognitive and affective) influence the genres of games they prefer/benefit from? (3) How can intelligent tutoring technologies augment gaming experience, with particular consideration to both motivational and learning outcomes? (4) How can we incorporate tutoring without interfering with game playing? (5) What role can intelligent educational games play in collaborative and social learning experiences? (6) The cost of developing games is very high, and adding AI techniques to the picture is likely to make the cost even higher. What tools exist or need to be developed to manage the development cost? (7) Should the gaming industry be involved and how?

By addressing these issues in an mixed-mode, informal set of interactions, this workshop will explore the feasibility and utility of Intelligent Educational Games, identify key problems to address, and contribute to advancing the state of the art of this emerging area of research.

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Motivation and Affect in Educational Software

Cristina Conati, University of British Columbia, Canada: conati@cs.ubc.ca
Benedict du Boulay, University of Sussex, UK: B.Du-Boulay@sussex.ac.uk
Claude Frasson, University of Montreal, Canada: frasson@iro.umontreal.ca
Lewis Johnson, USC, Information Sciences Institute, USA: johnson@isi.edu
Rosemary Luckin, University of Sussex, UK: R.H.Luckin@sussex.ac.uk
Erika A. Martinez-Miron, Univ. of Sussex, UK: E.A.Martinez-Miron@sussex.ac.uk
Helen Pain, University of Edinburgh, UK: helen@inf.ed.ac.uk
Kaska Porayska-Pomsta, University of Edinburgh, UK: kaska@inf.ed.ac.uk
Genaro Rebolledo-Mendez, Univ. of Sussex, UK: G.Rebolledo-Mendez@sussex.ac.uk

Motivation and affect (e.g., basic affective reactions such as like/dislike; specific emotions such as frustration, happiness, anger; moods; attitudes) often play an important role in learning situations. There have been various attempts to take them into account both at design time and at run time in AIED systems, though the evidence for the consequential impact on learning is not yet strong. Much research needs to be carried out in order to better understand this area. In particular, we need to deepen our knowledge of how affect and motivation relate to each other and to cognition, meta-cognition, learning context and teaching strategies/tactics. This workshop is intended bridge the gap existing between previous AIED research, particularly in motivation and meta-cognition, with the ever-increasing research in emotions and other affective components.

By bringing together researchers in the area, the workshop will be a forum to discuss different approaches with the aim of enriching our knowledge about how to create effective and affective learning environments. Also, it is expected to be a forum on which to address the appropriateness of defining bridges that could bring about new ways of relating cognitive and affective aspects of learning. At the end of the workshop we expect to reach agreements on which are the relevant emotions in learning contexts, as well as in the terminology been used so far (e.g. affect, emotion, motivation).

We invited papers, which present either finished, or work in progress or theoretical positions in the following areas:

- Affective/motivational modelling.
- Affective/motivational diagnosis.
- Relevant aspects of motivation and affect in learning.
- Strategies for motivational and affective reaction,
- Integrative models of cognition, motivation, and affect.
- Personal traits, motivation, and affect.
- Learning styles, learning domains and learning contexts.
- Learning goals, motivation, and affect.
- Influences of dialogues in affective computing.
- Use of agents as affective companions.
- Interface design for affective interactions.

The workshop is focused on exploring the following questions:

- Which emotions might be useful to model (e.g. basic affective reactions such as like/dislike; specific emotions such as frustration, happiness, anger; moods)?
- How do individual traits influence the learner's motivational state?
- How are motivation and emotional intelligence related?

The workshop is focused on exploring the following questions:

- Which emotions might be useful to model (e.g. basic affective reactions such as like/dislike; specific emotions such as frustration, happiness, anger; moods)?
- How do individual traits influence the learner's motivational state?
- How are motivation and emotional intelligence related?

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
© 2005 The authors. All rights reserved.

Third International Workshop on Authoring of Adaptive and Adaptable Educational Hypermedia

Dr. Alexandra Cristea - Eindhoven University of Technology, The Netherlands
Dr. Rosa M. Carro - University Autonoma of Madrid, Spain
Prof. Dr. Franca Garzotto - Politecnico di Milano, Italy

This workshop follows a successful series of workshops on the same topic. The current workshop focuses on the issues of design, implementation and evaluation of general Adaptive and Adaptable (Educational) Hypermedia, with special emphasis on the connection to user modelling and pedagogy. Authoring of Adaptive Hypermedia has been long considered as secondary to adaptive hypermedia delivery. This task is not trivial at all. There exist some approaches to help authors to build adaptive-hypermedia-based systems, yet there is a strong need of high-level approaches, formalisms and tools that support and facilitate the description of reusable adaptive websites. Only recently have we noticed a shift in interest, as it became clearer that the implementation-oriented approach would forever keep adaptive hypermedia away from the 'layman' author. The creator of adaptive hypermedia cannot be expected to know all facets of this process, but can be reasonably trusted to be an expert in one of them. It is therefore necessary to research and establish the components of an adaptive hypermedia system from an authoring perspective, catering for the different author personas that are required. This type of research has proven to lead to a modular view on the adaptive hypermedia. One of these modules, which is most frequently used, is the User Model, also called Learner Model in the Educational field (or Student Model in ITS). Less frequent, but also emerging as an important module is the Pedagogical Model (this model has also different names in different implementations, too various to name here). It becomes more and more clear that for Adaptive Educational Hypermedia it is necessary to consider not only the learner's characteristics, but also the pedagogical knowledge to deal with these characteristics. This workshop will cover all aspects of the authoring process of adaptive educational hypermedia, from design to evaluation, with special attention to Learner and Pedagogical models. Therefore, issues to discuss are:

- What are the main characteristics (that should be) modelled of learners?
- How can the pedagogical knowledge be formulated in a reusable manner?
- How can we consider user cognitive styles in adaptive hypermedia?
- How can we consider user learning styles in adaptive hypermedia?
- Are there any recurring patterns that can be detected in the authoring process generally speaking, and in the authoring of user or pedagogic model in particular?

The workshop will also lead to a better understanding and cross-dissemination of user-specific patterns extracted from existing design and authoring processes in AH, especially focused around user modelling and pedagogic modelling. The workshop aims to attract the interest of the related research communities to the important issues of design and authoring, with special focus on user and pedagogic models in adaptive hypermedia; to discuss the current state of the art in this field; and to identify new challenges in the field. Moreover, the workshop should be seen as a platform that enables the cooperation and exchange of information between European and non-European projects.

Major Themes of the workshop include:

- Design patterns for adaptive educational hypermedia
- Authoring user models for adaptive/adaptable educational hypermedia
- Authoring pedagogic models for adaptive/adaptable educational hypermedia

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

© 2005 The authors. All rights reserved.

Learner Modelling for Reflection, to Support Learner Control, Metacognition and Improved Communication between Teachers and Learners

Judy KAY[1], Andrew LUM[1] and Diego ZAPATA-RIVERA[2]
[1] *School of Information Technologies, University of Sydney, Australia.*
[2] *Educational Testing Service, Rosedale Road. Princeton, NJ 08541 USA*
{judy, alum}@it.usyd.edu.au, dzapata@ets.org

Learner modelling is at the core of AIED research, as the learner model is the foundation of 'systems that care' because they have the potential to treat learners as individuals. This workshop will bring together researchers working towards the many important, emerging roles for learner models. Personalising teaching is their core task. It is becoming increasingly clear that learner models are first class objects which can be made open to learners and teachers as a basis for improving learning outcomes. Essentially, open learner models offer the potential to help learners reflect on their own knowledge, misconceptions and learning processes.

A particularly important new direction is to incorporate open learner models into conventional learning systems. The challenge is to fruitfully make this data more useful as detailed models of learner development, with modelling of competence, knowledge and other aspects. A closely related area of importance is how best to collect, analyse and externalise data from learner interactions and how to represent this for most effective support of reflection. Another important new direction for open learner models is in the support of learner control over learning. At quite a different level, we are seeing the emergence of systems that model affective aspects such as emotion. We need to link this with the potential role of open learner models. Finally, there is considerable work in machine learning in conjunction with learner modelling. This is often predicated on the assumption that a machine learning system can access collections of student models.

Program committee:
Susan Bull, University of Birmingham, UK; Paul Brna, Northumbria University, UK; Peter Brusilovsky, University of Pittsburgh, USA; Al Corbett, Carnegie Mellon University, USA; Vania Dimitrova, University of Leeds, UK; Jim Greer, University of Saskatchewan, Canada; Gord McCalla, University of Saskatchewan, Canada; Rafael Morales, Northumbria University, UK; Kyparisia Papanikolaou, University of Athens, Greece; Nicolas Van Labeke, University of Nottingham, UK.

Workshop Chairs:
Judy Kay, University of Sydney, Australia
Andrew Lum, University of Sydney, Australia
Diego Zapata, Educational Testing Service, USA

This page intentionally left blank

Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005
C 2005 The authors. All rights reserved.

Author Index