Integrated Series in Information Systems 37

Series Editors: Ramesh Sharda · Stefan Voß

Jason Papathanasiou
Nikolaos Ploskas
Isabelle Linden *Editors*

Real-World Decision Support Systems

Case Studies

Springer

Integrated Series in Information Systems

Volume 37

Series editors

Ramesh Sharda
Oklahoma State University, Stillwater, OK, USA

Stefan Voß
University of Hamburg, Hamburg, Germany

More information about this series at http://www.springer.com/series/6157

Jason Papathanasiou • Nikolaos Ploskas •
Isabelle Linden
Editors

Real-World Decision Support Systems

Case Studies

Springer

Editors
Jason Papathanasiou
Dept. of Business Administration
University of Macedonia
Thessaloniki, Greece

Nikolaos Ploskas
Dept. of Chemical Engineering
Carnegie Mellon University
Pittsburgh, PA
USA

Isabelle Linden
Dept. of Business Administration
University of Namur
Namur, Belgium

ISSN 1571-0270 ISSN 2197-7968 (electronic)
Integrated Series in Information Systems
ISBN 978-3-319-43915-0 ISBN 978-3-319-43916-7 (eBook)
DOI 10.1007/978-3-319-43916-7

Library of Congress Control Number: 2016960674

© Springer International Publishing Switzerland 2016
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To the Euro Working Group on Decision Support Systems coordination board and members, for their continuous and inspiring commitment on the promotion of the discipline

Foreword

Decision support systems (DSSs) appeared in the literature by the beginning of the 1970s. The first developed DSS was developed for executive managers using personal computers and was called executive information systems. Since this period, DSS evolved in several directions. The first proposed architecture of these systems was composed by a database management system, a model base management system, and a man-machine interaction module. The first step in the evolution of DSS was based on the introduction of knowledge in the architecture. A new module was added called the knowledge-based management system as well as an inference engine. From then on, due to a huge amount of data, the database management system evolved in line with research on data warehouses, for which the main concern is to find suitable data for the decision-maker. For the model base management system, a lot of research has been conducted including several kinds of models of real decision problems. These models are formulated in different ways like linear or constraint programming, decision rules, decision trees, etc. Nowadays, researchers on DSS are still very active and dynamic, and we can notice an evolution of the name; DSSs are also called in a more general way decision-making support systems (DMSSs). The number of international journals and international conferences on this topic is progressing every day. Recently, a new such journal, the *International Journal of Decision Support System Technologies* was created, published by IGI Global. This journal publishes selected papers organized in one volume per year including four issues composed of four papers. We can also mention the International Conference on Decision Support System Technologies organized annually by the Euro Working Group on Decision Support Systems. The conference attracts every year an international group of researchers, academics, and practitioners working on decision support systems. Topics covered by both the journal and the conference are, among others, context awareness, modeling, and management for DMSS; data capture, storage, and retrieval; DMSS feedback control mechanisms; function integration strategies and mechanisms; DMSS network strategies and mechanisms; DMSS software algorithms; DMSS system and user dialog methods; system design, development, testing, and implementation;

DMSS technology evaluation; and finally DMSS technology organization and management.

Nevertheless, this research would be without any actual interest if applications would not be developed and tested in real-life situations. The applications of DSS or cases of DSS are also very important and allow researchers to implement their architectures, models, and methodologies in real situations. These implementations are very valuable for the improvement of the DSS field. Indeed, the idea of this book, *Real-World Decision Support Systems – Case Studies*, including the application domains of the environment, agriculture and forestry, business and finance, engineering, food industry, health, production and supply chain management, and urban planning, is an excellent initiative. Research on the DSS discipline is still very promising and will be exciting for several decades to come.

Toulouse, France — Pascale Zaraté
June 2016

Preface

The number of papers regarding decision support systems (DSSs) has soared during the recent years, especially with the advent of new technologies. Indeed, if someone considers DSS as an umbrella term [1], the plurality of research areas covered is striking: from computer science and artificial intelligence to mathematics and psychology [3]. It is in this context that the editors of this book felt that there is a gap in the overall fabric; it was felt that too much attention has been given to theoretical aspects and individual module design and development. In addition, there have been many failures in information systems development; poor initial requirements analysis and design has many times led to a notable lack of success. Indeed, it seems that the DSS discipline is rather prone to this, tagging the development of such projects as risky affairs [2].

Moreover, decisions today have to be made in a very complex, dynamic, and highly unpredictable international environment with various stakeholders, each with his own separate and sometimes hidden agenda. Right into the center of the whole decision process is the decision-maker; he has the responsibility for the final decision and he will most probably bear the consequences. As there is no model that can integrate all the possible variables that influence the final outcome and the DSS results have to be combined with the decision-maker's insights, background, and experience, the system must facilitate the process at each stage rendering the user experience concept of great significance.

Bearing the above in mind, the rationale behind this edition is to provide the reader with a set of cases of real-world DSS, as the book title suggests. The editors were interested in real applications that have been running for some time and as such tested in actual situations. And not only that; unsuccessful cases were targeted as well, systems that at some point of their life cycle were deemed as failures for one reason or another. If the systems failed, what were the (both implicit and explicit) reasons for that? How can they be recorded and avoided again? The lessons learned in both successful and unsuccessful cases are considered invaluable, especially if one considers the investment size of such projects [4]. The overall and primary goal in each case is to point out the best practices in each stage of the system life cycle, from the initial requirements analysis and design phases to the final stages of the project. The cases aim to stimulate the decision-makers and provide firsthand experiences, recommendations, and lessons learned so that failures can be avoided and successes can be repeated.

The authors of the chapters of this book were requested to provide information on a number of issues. They were asked to follow a certain chapter structure, and their work was rigorously peer-reviewed by the editors and selected reviewers from the DSS community. The cases are also presented in a constructive, coherent, and deductive manner, in order to act as showcases for instructive purposes, especially considering their high complexity. This book consists of one introductory chapter presenting the main concepts of a decision support system and 12 chapters that present real-world decision support systems from several domains. The first chapter by Daniel Power reviews frameworks for classifying and categorizing decision support systems, while it also addresses the need and usefulness of decision support system case studies.

Chapter 2 by Malik Al Qassas, Daniela Fogli, Massimiliano Giacomin, and Giovanni Guida presents the design, development, and experimentation of a knowledge-driven decision support system, which supports decision-making processes that occur during clinical discussions.

Chapter 3 by Anna Arigliano, Pierpaolo Caricato, Antonio Grieco, and Emanuela Guerriero proposes a method to integrate decision analysis techniques in high-throughput clinical analyzers. The proposed method is integrated into a clinical laboratory information system in order to demonstrate the benefits that it achieves.

Chapter 4 by Andrea Bettinelli, Angelo Gordini, Alessandra Laghi, Tiziano Parriani, Matteo Pozzi, and Daniele Vigo is about a suite of two decision support systems for tackling network design problems and energy-production management problems.

Chapter 5 by Pierpaolo Caricato, Doriana Gianfreda, and Antonio Grieco analyzes a model-driven decision support system to solve a variant of the cutting stock problem on a company that produces high-tech fabrics.

Chapter 6 by Mats Danielson, Love Ekenberg, Mattias Göthe, and Aron Larsson introduces a procurement decision support system implementing algorithms targeted for decision evaluation with imprecise data that it can be used as an instrument for a more meaningful procurement process.

Chapter 7 by António J. Falcão, Rita A. Ribeiro, Javad Jassbi, Samantha Lavender, Enguerran Boissier, and Fabrice Brito presents a model-driven evaluation support system for open competitions within Earth observation topics.

Chapter 8 by Narain Gupta and Goutam Dutta presents the design, development, and implementation of a model-based decision support system for strategic planning in process industries.

Chapter 9 by Andreja Jonoski and Abdulkarim H. Seid explains the experiences in developing and applying a model-driven decision support system in a transboundary river basin context, taking the Nile Basin decision support system as a case.

Chapter 10 by Manfred J. Lexer and Harald Vacik presents a data-driven decision support system for forest management that can support all phases of the decision-making process.

Chapters 11 and 12 by Mário Simões-Marques examine in detail a decision support system for emergency management. Chapter 11 describes the problem context, the system requirements and architecture, the knowledge management process, and the spiral development approach, while Chap. 12 presents the main features implemented in the proposed decision support system.

Finally, Chap. 13 by Mette Sønderskov, Per Rydahl, Ole M. Bøjer, Jens Erik Jensen, and Per Kudsk presents a knowledge-driven decision support system for weed control that offers herbicide dose suggestions based on a large database of the existing knowledge of herbicides and herbicide efficacies.

We are very delighted to have included in this book a set of high-quality and interesting pieces of research, authored by researchers and industrial partners coming from different research institutions, universities, and companies across different continents. We are grateful to all reviewers and authors for the collaboration and work they have put into this book. We especially want to thank Daniel Power for writing the introductory chapter that introduces the main concepts that define a decision support system and prepares the readers for the remaining chapters of this book.

We hope that you will also enjoy reading the book, and we hope the presented "good" and "bad" practices on developing and using a decision support system can be useful for your research.

Thessaloniki, Greece — Jason Papathanasiou
Pittsburgh, PA, USA — Nikolaos Ploskas
Namur, Belgium — Isabelle Linden

References

1. Alter, S.: A work system view of DSS in its fourth decade. Decis. Support Syst. **38**(3), 319–327 (2004)
2. Arnott, D., Dodson, G.: Decision support systems failure. In: Burstein, F., Holsapple, C.W. (eds.) Handbook on Decision Support Systems 1 Basic Themes, pp. 763–790. Springer, Berlin, Heidelberg (2008)
3. Eom, S.B.: Reference disciplines of decision support systems. In: Burstein, F., Holsapple, C.W. (eds.) Handbook on Decision Support Systems 1 Basic Themes, pp. 141–159. Springer, Berlin, Heidelberg (2008)
4. Prakken, B.: The (economic) evaluation of investments in information systems and in ICT. In: Prakken, B. (ed.) Information, Organization and Information Systems Design, pp 197–222. Springer, Berlin (2000)

Contents

Contributors

Malik Al Qassas University of Brescia, Brescia, Italy

Anna Arigliano DII – Università del Salento, Via Monteroni, Lecce, Italy

Andrea Bettinelli OPTIT Srl, Viale Amendola, Imola, Italy

Enguerran Boissier Terradue UK Ltd, Old Jewry, London, UK

Ole M. Bøjer IPM Consult Ltd, Hovedgaden, Stenlille, Denmark

Fabrice Brito Terradue UK Ltd, Old Jewry, London, UK

Pierpaolo Caricato DII – Università del Salento, Via Monteroni, Lecce, Italy

Mats Danielson Department of Computer and Systems Sciences, Stockholm University, Kista, Sweden

International Institute for Applied Systems Analysis, Laxenburg, Austria

Goutam Dutta Indian Institute of Management, Ahmedabad, India

Love Ekenberg International Institute for Applied Systems Analysis, Laxenburg, Austria

Department of Computer and Systems Sciences, Stockholm University, Kista, Sweden

António J. Falcão UNINOVA, Campus FCT/UNL, Monte da Caparica, Portugal

Daniela Fogli University of Brescia, Brescia, Italy

Massimiliano Giacomin University of Brescia, Brescia, Italy

Doriana Gianfreda DII – Università del Salento, Via Monteroni, Lecce, Italy

Angelo Gordini OPTIT Srl, Viale Amendola, Imola, Italy

Mattias Göthe Department of Computer and Systems Sciences, Stockholm University, Kista, Sweden

Antonio Grieco DII – Università del Salento, Via Monteroni, Lecce, Italy

Emanuela Guerriero DII – Università del Salento, Via Monteroni, Lecce, Italy

Giovanni Guida University of Brescia, Brescia, Italy

Narain Gupta Management Development Institute, Gurgaon, Haryana, India

Javad Jassbi UNINOVA, Campus FCT/UNL, Monte da Caparica, Portugal

Jens Erik Jensen SEGES P/S, Aarhus N, Denmark

Andreja Jonoski UNESCO-IHE Institute for Water Education, DA Delft, The Netherlands

Per Kudsk Department of Agroecology, Aarhus University, Forsøgsvej, Slagelse, Denmark

Alessandra Laghi OPTIT Srl, Viale Amendola, Imola, Italy

Department of Electrical, Electronic and Information Engineering "G. Marconi", Viale Risorgimento, Bologna, Italy

Aron Larsson Department of Computer and Systems Sciences, Stockholm University, Kista, Sweden

Department of Information and Communications Systems, Mid Sweden University, Sundsvall, Sweden

Samantha Lavender Pixalytics Ltd, Derriford, Plymouth, Devon, UK

Manfred J. Lexer University of Natural Resources and Life Sciences, Vienna, Austria

Tiziano Parriani OPTIT Srl, Viale Amendola, Imola, Italy

Daniel J. Power University of Northern Iowa, Cedar Falls, IA, USA

Matteo Pozzi OPTIT Srl, Viale Amendola, Imola, Italy

Rita A. Ribeiro UNINOVA, Campus FCT/UNL, Monte da Caparica, Portugal

Per Rydahl IPM Consult Ltd, Hovedgaden, Stenlille, Denmark

Abdulkarim H. Seid NBI Secretariat, Entebbe, Uganda

Mário Simões-Marques Centro de Investigação Naval (CINAV) – Portuguese Navy, Alfeite, Almada, Portugal

Mette Sønderskov Department of Agroecology, Aarhus University, Forsøgsvej, Slagelse, Denmark

Harald Vacik University of Natural Resources and Life Sciences, Vienna, Austria

Daniele Vigo OPTIT Srl, Viale Amendola, Imola, Italy

Department of Electrical, Electronic and Information Engineering "G. Marconi", Viale Risorgimento, Bologna, Italy

List of Reviewers

Guy Camilleri, IRIT, France
Csaba Csáki, University College Cork, Ireland
Pavlos Delias, Technological Institute of Kavala, Greece
Themistoklis Glavelis, University of Macedonia, Greece
Isabelle Linden, University of Namur, Belgium
Jason Papathanasiou, University of Macedonia, Greece
Nikolaos Ploskas, Carnegie Mellon University, USA
Theodore Tarnanidis, University of Macedonia, Greece
Georgios Tsaples, University of Rome, Italy
Andy Wong, University of Strathclyde, UK

The editors of this book wish to acknowledge their gratitude for the prompt and highly constructive reviews received from the researchers above in the various phases of this book's reviewing process.

About the Editors

Jason Papathanasiou is an assistant professor at the Department of Business Administration, University of Macedonia, Greece. His PhD was in operational research and informatics and he has worked for a number of years at various institutes. He has organized and participated in many international scientific conferences and workshops. He has published more than 100 papers in international peer-referred journals, conferences, and edited volumes and has participated in various research projects in FP6, FP7, Interreg, and COST; he served also as a member of the TDP Panel of COST and currently serves at the coordination board of the EURO Working Group of Decision Support Systems. His research interests include decision support systems, operational research, and multicriteria decision-making.

Nikolaos Ploskas is a postdoctoral researcher at the Department of Chemical Engineering, Carnegie Mellon University, USA. His primary research interests are in operations research, decision support systems, mathematical programming, linear programming, and parallel programming. He has participated in several international and national research projects. He is author of more than 50 publications in high-impact journals, book chapters, and conferences. He has also served as reviewer in many scientific journals. He was awarded with an honorary award from HELORS (HELlenic Operations Research Society) for the best doctoral dissertation in Operations Research (2014).

Isabelle Linden is a professor of information management at the University of Namur in Belgium, Department of Business Administration. She obtained her PhD in computer sciences from the University of Namur. She also holds masters degrees in philosophy and in mathematics from the University of Liège, Belgium. She is member of the CoordiNam Laboratory and the FoCuS Research Group. Combining theoretical computer science and business administration, her main research domain regards information, knowledge, and artificial intelligence. She explores their integration within systems as EIS, DSS, and BI systems. Her works can be found in several international edited books, journals, books chapters, and conferences. She serves as reviewer and program committee member in several international journals, conferences, and workshops.

Chapter 1
Computerized Decision Support Case Study Research: Concepts and Suggestions

Daniel J. Power

Abstract Supporting decision making is an important and potentially transformative research topic that is challenging for academic researchers to study. Anecdotal evidence suggests that computerized decision support systems (DSS) can improve decision quality and change the structure and functioning of organizations. To make progress in our understanding of this phenomenon there is an ongoing need for more decision support case study field research that includes documenting decision support impacts. Research case studies help understand the use and consequences associated with building and using computerized decision support. More descriptive and technical information about specific DSS will be helpful in explaining the variability of these technology artifacts. Current theory related to computerized decision support is inadequate and research case studies can potentially assist in theory building. The possibilities for improving and increasing decision support continue to evolve rapidly and research case studies can help define this expanding, changing field of study. More "good" case studies and more details about each specific case is useful, helpful, and a significant contribution to understanding how computing technologies can improve human decision making.

1.1 Introduction

A variety of tools and aids have been used by people to help make decisions for thousands of years. For example, people have kept ledgers and records of historical information, have used checklists and have built physical scale models. Now managers use these tools and more sophisticated computerized tools for decision support. Computerized decision support systems and analytics can serve many new purposes and are and will be built using many differing technologies. The domain of computerized decision support continues to get more diverse and more sophisticated.

D.J. Power (✉)
University of Northern Iowa, Cedar Falls, IA 50614-0125, USA
e-mail: Daniel.Power@uni.edu

© Springer International Publishing Switzerland 2016

J. Papathanasiou et al. (eds.), *Real-World Decision Support Systems*,
Integrated Series in Information Systems 37,
DOI 10.1007/978-3-319-43916-7_1

Decision support capabilities should have a targeted user group and a purpose. A decision support system is a technology artifact crafted from hardware and software linked by networks and accessed by interface devices like smart phones and personal computers. Documenting the expanding application domain is a major reason to prepare research case studies. DSS builders must remember that providing computerized decision support does not guarantee that better decisions will be made. Understanding and documenting Decision Support Systems (DSS) can potentially improve the design and usefulness of DSS. This chapter focuses on using case study research to understand computerized decision support. This chapter reviews the ongoing need for case study field research and documenting UML use cases related to decision support. Section 1.2 reviews frameworks for classifying and categorizing computerized DSS. Section 1.3 reviews the case study method in general and then discusses applying the method to documenting a specific DSS artifact or to examining a DSS in its context of application and use. Section 1.4 reviews classical DSS case studies. Section 1.5 addresses the usefulness of DSS case studies. Section 1.6 summarizes major conclusions from this methodology overview and some recommendations for using a case study to study computerized decision support.

1.2 Understanding Decision Support Systems

At the website DSSResources.com, a decision support system (DSS) is defined as "an interactive computer-based system or subsystem intended to help decision makers use communications technologies, data, documents, knowledge and/or models to identify and solve problems, complete decision process tasks, and make decisions. Decision support system is a general term for any computer application that enhances a person or group's ability to make decisions. In general, decision support systems are a class of computerized information systems that support decision-making activities."

Decision support is a broad concept that describes tools and capabilities to assist individuals, groups, teams and organizations during decision making processes. Computerized decision support systems built since the 1950s can be categorized in a number of ways, cf. [24]. The four major taxonomies or frameworks in the literature were proposed by Alter [1], Arnott and Pervan [2], Holsapple and Whinston [9], and Power [20–22, 28]. There are commonalities among them and the schemes are not contradictory. All of the frameworks attempt to organize observations and literature about the variety of DSS that have been built and used over the years. This review focuses on Power's [20, 21] expanded DSS framework that builds upon Alter's [1] categories.

There are five DSS types in the expanded framework defined based upon the dominant technology component. The initial DSS category in the expanded framework is model-based or model-driven DSS. Many early DSS derived their functionality from quantitative models and limited amounts of data. Scott-Morton's

[33] production planning management decision system was the first widely discussed model-driven DSS. Early case studies of other model-driven systems were about MEDIAC [15], SPRINTER [40] and BRAND AID [14]. A model-driven DSS emphasizes access to and manipulation of financial, optimization and/or simulation models. Simple quantitative models provide the most elementary level of functionality. Model-driven DSS generally use small to medium-sized data sets, and parameters are often provided by decision makers. These systems aid decision makers in analyzing a situation and evaluating sensitivity issues, but in general large, gigabyte or terabyte data bases are not needed for model-driven DSS, cf. [21].

Alter [1] identified data-oriented DSS as fundamentally different than DSS deriving functionality more from quantitative models than from data. Data sets were growing, but analytical tools were limited. Bonczek, Holsapple and Whinston [4] termed these systems retrieval-only DSS. Data-driven DSS emphasize access to and manipulation of large data sets. Simple online file systems accessed by query and retrieval tools provide the most elementary level of functionality. Data warehouse and Business Intelligence systems that provide for the manipulation of data by computerized tools provide additional functionality.

Beginning in the mid-1970s the developments in Artificial Intelligence led to creating knowledge-driven DSS. These systems suggest or recommend actions. Alter [1] termed them suggestion DSS and Klein and Methlie [13] used the term knowledge-based DSS. These knowledge-driven DSS are person-computer systems with specialized problem-solving expertise.

Two remaining categories in the expanded DSS framework [19, 21] are communications-driven and document-driven DSS. Communications-driven DSS "use network and communications technologies to facilitate decision-relevant collaboration and communication. In these systems, communication technologies are the dominant architectural component. Tools used include groupware, video conferencing and computer-based bulletin boards" [21]. A document-driven DSS "uses computer storage and processing technologies to provide document retrieval and analysis. Large document databases may include scanned documents, hypertext documents, images, sounds and video. Examples of documents that might be accessed by a document-driven DSS are policies and procedures, product specifications, catalogs, and corporate historical documents, including minutes of meetings and correspondence. A search engine is a primary decision-aiding tool associated with a document-driven DSS" [21]. Table 1.1 provides examples of the dimensions in the expanded framework.

The expanded framework identifies the primary dimension for categorizing DSS is the dominant architecture technology component or driver that provides decision support. The three secondary dimensions are the targeted users, the specific purpose of the system and the primary deployment or enabling technology. Five generic DSS types are identified and defined based upon the dominant technology component. This framework is the conceptualization used at DSSResources.COM to organize what we have learned about decision support systems, cf. [19, 23]. Table 1.2 provides a general checklist for categorizing the five broad types of decision support systems.

Table 1.1 Expanded DSS framework [25]

DSS type	Dominant DSS component	Targeted users (examples)	Purpose (examples)	Enabling technology (examples)
Communications-driven DSS	Communications	Internal teams	Conduct a meeting	Bulletin board
		Supply chain partners	Help users collaborate	Videoconferencing
Data-driven DSS	Database	Managers and staff, now suppliers	Query a data warehouse	Relational databases
				Multidimensional databases
Document-driven DSS	Document storage and management	Specialists and user group is expanding	Search Web pages	Search engines, HTML
Knowledge-driven DSS	Knowledge base, AI	Internal users, new customers	Management advice	Expert Systems
Model-driven DSS	Quantitative models	Managers and staff, new customers	Scheduling	Linear Programming, Excel
			Forecasting	

Table 1.2 Check list for categorizing decision support systems

DSS check list
1. What is the dominant component of the architecture that provides functionality?
2. Who are the targeted users?
3. What is the purpose of the DSS?
4. What enabling technology is used to deploy the DSS?

1.3 Decision Support Case Studies

A case study is one type of qualitative research method. A case study researcher often uses both observation and systematic investigation to gather data and then the case write-up documents and summarizes what was found. Ideally a researcher needs access to observe the decision support capability in use, access to documents, and also access to ask questions of both developers and users.

Case studies help us understand computerized decision support. Both teaching and research case studies serve a useful purpose in advancing the field. A good teaching case can share challenges faced in design, implementation, and use. A good research case study can generate hypotheses for further testing and document "best practices" and use cases. Even short case study examples and vendor reported case studies enrich our courses and help explain the breadth of the decision support phenomenon.

In general, a research case study presents a systematic description, explanation and analysis of a specific instance of a category or sub-category of objects or artifacts. Decision support artifacts are especially important to study. Software systems can vary greatly, and each specific artifact we investigate informs our understanding of what is possible, what has worked and been effective, and what might work in a different context.

Schell [32] argues "As a form of research, the case study is unparalleled for its ability to consider a single or complex research question within an environment rich with contextual variables". He defines three characteristics of an empirical or research case study: (1) investigates a contemporary phenomenon within its real-life context; (2) the boundaries between phenomenon and context are not clearly evident; and (3) multiple sources of evidence are used, cf. [44].

Wikipedia.com notes "A case study involves an up-close, in-depth, and detailed examination of a subject (the case), as well as its related contextual conditions." (cf. https://en.wikipedia.org/wiki/Case_study). In general, decision support case studies should be "key" cases that are chosen because of the inherent interest of the case or the circumstances surrounding it.

WhatIs.com defines a case study in a business context as "a report of an organization's implementation of something, such as a practice, a product, a system or a service. The case study can be thought of as a real-world test of how the implementation works, and how well it works" [42].

Case studies are a form of qualitative descriptive research. An ongoing concern are the issues of validity, reliability, and generalizability, cf. [7]. In most situations it is desirable to use several methods of data collection including observing people using the system, structured feedback from users, review of technical documentation, etc. Case studies based on multiple sources of information are often perceived as more valid and reliable.

A Google search on the key words "decision support case study" in quotations suggests the case study is a reasonably popular research method for this decision support phenomenon. The actual search in November 2015 returned 2330 results. Without using quotations around the phrase the search returned about 43 million results. Cases studies were identified that reported systems serving many diverse purposes including: clinical decision support (CDS), risk management, capacity planning, flood forecasting, technology selection, veterinary decision support, investments, land use planning, and scheduling to name a few of them.

Can we generalize from an individual case study or even 2330 case studies? Generalization can result from examining specific case studies, but the credibility of the generalization increases as more cases are examined. Decision support case studies provide a description of a software artifact and its context of use, and an implementation case study can identify what did not work and sometimes reasons why failure occurred. A case study can also help identify design patterns and best practices in terms of design methods, implementation processes, and deployment and ongoing use of a decision support capability. Also, case studies of the same or different systems at various stages in the software life cycle can help piece together the longitudinal interaction of software systems and decision makers. So we may be able to develop useful generalizations from case study findings.

Decision support case studies are important because "good" ones provide detailed information about how software/hardware systems are impacting decision making in an actual organization. The decision support phenomenon becomes more concrete and the rich context can be shared along with technical details and observational notes.

1.4 Examples of DSS Case Studies

DSS case studies published in journals and books have contributed significantly to our understanding. Websites like DSSResources.com and vendor websites also include case examples. To document his framework, Alter [1] explained eight major case examples, Connoisseur Foods, Great Eastern Bank OPM, Gotaas-Larse Shipping Corporate Planning System, Equitable Life Computer-Assisted Underwriting System, a media decision support system, Great Northern Bank budgeting, planning and control system, Cost of Living Council DSS, and AAIMS, an analytical information system.

A common motivation for adopting or building information systems and decision support systems is that the organization will gain a competitive advantage. There is some case study evidence to support that claim. For example, in a literature review, Kettinger, Grover, Guha, and Segars [11] identified a number of companies that had gained an advantage from information systems and some of those systems were decision support systems. They identified nine case studies of DSS including:

1. Air Products—vehicle scheduling system
2. Cigna—health risk assessment system
3. DEC—expert system for computer configuration
4. First National Bank—asset management system
5. IBM—marketing management system
6. McGraw Hill—marketing system
7. Merrill Lynch—cash management accounts
8. Owens-Corning—materials selection system
9. Proctor & Gamble—customer response system

Power [21] explored the question of gaining competitive advantage from DSS by reviewing examples of decision support systems that provided a competitive advantage including systems at Frito-Lay, L.L. Bean, Lockheed-Georgia, Wal-Mart and Mrs. Field's Cookies. A major lesson learned from reviewing case studies is that a company needs to continually invest in a Strategic DSS to maintain any advantage.

Power [26, 27] identified classic Decision Support Systems described in case studies. A classic decision support system is an early and lasting example of using technology to support decision making. Ten DSS related to business and organization decision-making are among the classics: AAIMS, Advanced Scout, CATD, DELTA, Flagstar LIVE, GADS, GroupSystems, OPM, PMS and PROJECTOR.

The classic DSS help document what was possible even though the purpose of the systems may have been implemented using new technologies.

AAIMS, An Analytical Information Management System, was implemented by American Airlines in the mid-1970s. It was developed in APL and was used for data analysis. AAIMS included a database and functions for data retrieval, manipulation and report generation. The database included sales, price and employee data. Klass and Weiss developed the system internally at American Airlines. The system was used for ad hoc reporting and to create a report of corporate performance indicators, cf. [1, 12, 39].

Advanced Scout was developed by IBM and the software used data mining to help National Basketball Association (NBA) coaches and league officials organize and interpret data collected at every game. In the 1995–1996 season, 16 of 29 teams used the DSS. A coach can quickly review countless statistics: shots attempted, shots blocked, assists made, personal fouls. But Advanced Scout can also detect patterns in these statistics that a coach may not have identified. Patterns found through data mining are linked to the video of the game. This lets a coach look at just those video clips that make up the interesting pattern, cf. [3].

CATD or Computer Aided Train Dispatching was developed by the Southern Railway Co. from 1975 to 1982. It was initially built as a mini-computer based simulator and was installed and tested on the North Alabama track system in January 1980. The system was placed in production for that system on September 15, 1980. Gradually additional track systems were converted to CATD. The system provides decision support to aid train dispatchers in centralized traffic control. The system significantly reduced delays and reduced train meetings in the system, cf. [31].

DELTA, Diesel-Electric Locomotive Troubleshooting Aid, helped maintenance personnel to identify and correct malfunctions in diesel electric locomotives by applying diagnostic strategies for locomotive maintenance. The system can lead the user through a repair procedure. It was a rule-based system developed in a general-purpose representation language written in LISP. DELTA accesses its rules through both forward and backward chaining and uses certainty factors to handle uncertain rule premises. Although the system was prototyped in LISP, it was later reimplemented in FORTH for installation on microprocessor-based systems. The General Electric Company developed this system at their research and development center in Schenectady, New York. Current status unknown, but it was field tested, cf. [41].

Flagstar Bank, FSB (Nasdaq:FLGS) won the 1997 Computerworld Smithsonian Award for it's use of information technology in the Finance, Insurance, and Real Estate category. Flagstar Banks Lenders' Interactive Video Exchange (LIVE) merged Intel ProShare conferencing systems with automated underwriting technologies to allow the home buyer and loan underwriter to meet face to face and get loans approved quickly, regardless of where the loan originated. Usually this process takes weeks and the prospective home owner has no contact with the person who actually makes the decision, cf. [6].

GADS was an interactive system also known as Geodata Analysis and Display System. The goal in developing GADS was to enable nonprogrammers to solve unstructured problems more effectively by applying their job-specific experience and their own heuristics. It had a strong graphic display and “user-friendly” characteristics that enabled non-computer users to access, display, and analyze data that have geographic content and meaning. The system was used initially by police officers to analyze data on “calls for service”. By 1982, 17 specific DSS had been developed using GADS, cf. [37].

In early 1987, IBM combined efforts with the University of Arizona to implement a group decision support system (GDSS) called GroupSystems. GroupSystems was the result of a research and prototype development project by the MIS department. GroupSystems utilized a set of flexible software tools within a local area network to facilitate problem-solving techniques including brainstorming, idea organization, alternative generation, and alternative selection. The GroupSystems hardware, software and methodologies are combined in specially developed group facilities called decision support centers (DSC). These rooms were 26 feet by 30 feet and contained 11 PCs connected by a LAN to a large screen projector. The PC workstations were placed in a U-shape around the screen, cf. [16].

OPM, On-line Portfolio Management System, was described in a case study written by Alter [1] based on research done by Ginzberg. “OPM had four purposes: investment decision making, account reviews, administration and client relations, and training (p. 29)”. OPM included 8 functions: directory, scan, groups, table, histogram, scatter, summary and issue.

PMS, Portfolio Management System, was developed by T.P. Gerrity and it was implemented in four banks beginning in 1974. The purpose of the DSS was to help manage security portfolios and manage risk and return, cf. [10].

Finally, PROJECTOR was developed in 1970 by C.L. Meador and D.N. Ness to support financial planning. The system included forecasting and optimization models. It was used in 1974 by a New England manufacturing company to investigate the acquisition of a new subsidiary, cf. [17].

Based upon available descriptions the classic DSS can be classified as follows: AAIMS, OPM and PMS are data-driven DSS. GADS is a data-driven, spatial DSS. CATD is a model-driven DSS. DELTA is a knowledge-driven DSS. GroupSystems is a model-driven, group DSS.

At DSSResources.com, there are 54 case studies posted primarily between 2001 and 2007. There are examples of each of the five categories of decision support systems. The Decision Support Case Studies web page.is at URL http://dssresources.com/cases/index.html. The page preface notes: This DSSResources.com page indexes case examples of various types of computerized Decision Support Systems, decision automation systems and special decision support studies that use computerized analyses. Some of the cases are based upon field research, but many have been provided by software vendors. We have tried to confirm and verify the information in vendor supplied cases. The following examples from the case studies index are grouped into the five categories in the expanded framework.

Data-driven DSS. The Databeacon East of England Observatory case is a web-based system. The MySQL Cox Communications case describes an open source data-driven DSS for real-time operational support and management control. Stevens describes implementing the Redland Genstar Data Mart. Power and Roth describe Ertl's Decision Support Journey. Power documents GE Real Estate's Innovative Data-driven Decision Support.

Model-driven DSS. Stottler Henke Associates described PADAL a DSS that helps US Navy aircraft land aboard carriers. Procter & Gamble used @RISK and PrecisionTree. TechComm Associates documented how estimating software yielded higher profits at Liberty Brass. ProModel reported how MeritCare Health System used simulation to optimize integration of service areas into a new day unit.

Knowledge-driven DSS. Biss wrote about how Dynasty Triage Advisor enabled Medical Decision Support. Pontz and Power describe building an Expert Assistance System for Examiners (EASE) at the Pennsylvania Department of Labor and Industry. EXSYS reported how IAP Systems is using Exsys CORVID expert system software to support corporate families on overseas assignments.

Communications-driven DSS. eRoom staff documented how Naval Medicine CIOs used a collaboration and knowledge-sharing decision support application.

Document-driven DSS. Documentum Staff explained how BFGoodrich (BFG) Aerospace was improving Aircraft Maintenance Operations decision making using a document-driven work flow system. Stellent Staff reported how University of Alberta increased access to policies and procedures.

Some systems have multiple decision support subsystems. For example, Tully explains E-Docs Asset GIS, a web-based spatial DSS with both data and document-driven decision support.

1.5 How Useful Are DSS Case Studies

Decision Support Systems (DSS) encompass a broad array of software artifacts intended to support decision making. The broad purpose is the same for all DSS, but the narrower more specific uses and purposes vary. The targeted users of the systems also differ. More fundamentally the architecture, technologies and source of primary functionality can differ in significant ways. To better understand the wide range of systems categorized broadly as Decision Support Systems researchers can and should investigate exemplar systems and document them to demonstrate changes as DSS are built using new technologies and to document innovation and best practices.

The specific DSS in a specific context is the "case" being studied and researchers need to exercise care to insure their investigation does not bias the data collection or the analysis. A researcher collecting data about the design, functioning and effectiveness of a specific decision support system may and often is biased toward the expanded use of computerized decision support. Yin [44] defines the case study research method as an empirical inquiry that investigates a contemporary phenomenon within its real-life context (cf. p. 23). Prospective DSS case study researchers should consult sources like Soy [36], who suggests steps for preparing a case study for technology artifacts. He based his prescriptions on [35, 38, 44].

According to Yin [44–46], case studies are appropriate when your research addresses either a descriptive question like 'What is happening or has happened?' or an explanatory question like 'How or why did something happen?' Eisenhardt [5] concludes theory-building case study methods are appropriate for new areas of research as well as "research areas for which existing theory seems inadequate" (p. 549).

Some decision support case studies are longitudinal involving repeated observation and data collection over time while others involve a cross-sectional snapshot of the system. Both approaches have advantage and can potentially provide differing insights and different types of evidence. Selecting a specific DSS to study is most often based upon opportunity, cooperation of the "owner" of the DSS, and interest of the researcher or research team.

A systematic, research case study is in many ways the most useful research method for understanding the what, how, why and how much benefit questions important in an applied scientific field like computerized decision support. Reporting the implementation of a novel DSS is also useful, but some third party validation is desirable.

More case studies of Decision Support Systems in use are needed to improve our understanding and to document what is occurring. More longitudinal case studies that report design, development, installation, use, and maintenance would also be useful. Case studies provide rich, detailed information. DSS case study research is not often theory driven, it is not hypothesis testing, and a single case study does not result in generalizations, but it is useful. DSS case study research at its best leads to informed descriptions and interpretive theory development. Peskin [18] notes good description provides a foundation for all research. He also states "Interpretation not only engenders new concepts but also elaborates existing ones (p. 24)."

1.6 Conclusions and Recommendations

The value of a decision support case study depends upon many factors. Only some of them are controllable by the researchers. The following suggestions should increase the value of a DSS research case study and help to expand our collective body of decision support knowledge:

1. Try to identify novel DSS implementations where permission to publish the findings is granted.
2. Identify installations/sites where you receive good cooperation from both users and technical staff.
3. Be systematic in gathering information; think about what you want to know and what has been reported in other DSS case studies.
4. Try to use the actual decision support system. If possible, do more than observe its use.

5. Identify multiple informants and information sources, including system documentation.
6. Take notes, lots of notes.
7. Follow up a site visit or online meeting/demonstration with emails to get more details and to confirm what you heard and observed.
8. Say thank you often. Maintain positive relationships so you can get feedback on the draft of the case study. Make sure managers recognize the value of documenting the DSS, and of its development and use.

Yin [44] notes “The detective role offers some rich insights into case study field research (p. 58).” Like a detective, the case study researcher must know the purpose of the investigation, collect descriptive and factual data systematically, interpret the data, summarize what was found, and draw reasonable conclusions. Simon [34] briefly discussed the case study as an example of descriptive research. He admonishes the case study researcher to “work objectively. Describe what is really out in the world and what could be seen by another observer. Avoid filtering what you see through the subjective lenses of your own personality (pp. 276–277).”

Case study research is a legitimate tool for expanding our understanding of computerized decision support [8, 43]. No research methodology answers all of our questions conclusively. Qualitative DSS case study research brings an information systems researcher in direct contact with the technology artifact. The benefit to the researcher from that direct contact is enhanced by spending the time and effort to systematically collect data, organize and interpret the findings, and then share the case study with other researchers. Decision support researchers need to study in the field the decision support systems that they teach about, find interesting, and perhaps wonder about. Decision support systems are varied, complex, changing and consequential, some are more enduring than others. More research case studies and more details about each specific case will be useful, helpful, and a contribution to our understanding how computing and software can improve individual, group and organization decision making.

Note

This chapter incorporates material from Ask Dan! columns written by D. Power that have appeared in Decision Support News. Check [23, 29, 30] in the archive of Ask Dan! columns at http://dssresources.com. Thanks to Professor Dale Cyphert and the editors for suggestions.

References

1. Alter, S.L.: Decision Support Systems: Current Practice and Continuing Challenge. Addison-Wesley, Reading, MA (1980)
2. Arnott, D., Pervan, G.: A critical analysis of decision support systems research. J. Inf. Technol. **20**(2), 67–87 (2005)
3. Bhandari, I., Colet, E., Parker, J., Pines, Z., Pratap, R., Ramanujam, K.: Advanced Scout: data mining and knowledge discovery in NBA data. Data Min. Knowl. Disc. **1**(1), 121–125 (1997)
4. Bonczek, R.H., Holsapple, C.W., Whinston, A.: Foundations of Decision Support Systems. Academic, New York (1981)
5. Eisenhardt, K.M.: Building theories from case study research. Acad. Manag. Rev. **14**(4), 532–550 (1989)
6. Flagstar Bank: FSB wins 1997 Computerworld Smithsonian Award (1997). Available at http://www.thefreelibrary.com/Flagstar+Bank,+FSB+wins+1997+Computerworld+Smithsonian+Award-a019484611
7. Flyvbjerg, B.: Five misunderstandings about case-study research. Qual. Inq. **12**(2), 219–245 (2006)
8. Gummesson, E.: Qualitative Methods in Management Research. Sage, Newbury Park (1991)
9. Holsapple, C.W., Whinston, A.B.: Decision Support Systems: A Knowledge-based Approach. West Publishing Co., Minneapolis, MN (1996)
10. Keen, P.G.W., Scott-Morton, M.S.: Decision Support Systems: An Organizational Perspective. Addison-Wesley, Reading, MA (1978)
11. Kettinger, W.J., Grover, V., Guha, S., Segars, A.H.: Strategic information systems revisited: a study in sustainability and performance. MIS Q. **18**(1), 31–58 (1994)
12. Klass, R.L.: A DSS for airline management. In: ACM SIGMIS Database. Proceedings of a Conference on Decision Support Systems, Santa Clara, CA, vol. 8, no. 3, pp. 3–8 (Winter 1977)
13. Klein, M., Methlie, L.B.: Knowledge-Based Decision Support Systems with Applications in Business. Wiley, Chichester (1995)
14. Little, J.D.C.: Brandaid, an on-line marketing mix model, part 2: implementation, calibration and case study. Oper. Res. **23**(4), 656–673 (1975)
15. Little, J.D.C., Lodish, L.M.: A media planning calculus. Oper. Res. **17**, 1–35 (1969)
16. McGoff, C., Hunt, A., Vogel D., Nunamaker, J.: IBM's experiences with GroupSystems. Interfaces **20**(6), 39–52 (1990)
17. Meador, C.L., Ness, D.N.: Decision support systems: an application to corporate planning. Sloan Manag. Rev. **15**(2), 51–68 (1974)
18. Peskin, A.: The goodness of qualitative research. Educ. Res. **22**(2), 23–29 (1993)
19. Power, D.J.: Web-based and model-driven decision support systems: concepts and issues. In: Proceedings of the 2000 Americas Conference on Information Systems, Long Beach, CA (2000)
20. Power, D.J.: Supporting decision-makers: an expanded framework. In: Harriger, A. (ed.) Proceedings of the 2001 Informing Science Conference e-book Proceedings, Krakow, Poland, pp. 431–436 (2001)
21. Power, D.J.: Decision Support Systems: Concepts and Resources for Managers. Greenwood/Quorum, Westport, CT (2002)
22. Power, D.J.: Specifying an expanded framework for classifying and describing decision support systems. Commun. Assoc. Inf. Syst. **13**(1), 52 (2004)
23. Power, D.J.: What is the expanded DSS framework? DSS News (2004). Available at http://dssresources.com/faq/index.php?action=artikel&id=2
24. Power, D.J.: A brief history of decision support systems (2007). Available at http://DSSResources.COM/history/dsshistory.html
25. Power, D.J.: Decision Support Basics. Business Expert Press, New York (2009)

26. Power, D.J.: What are some classic business DSS? DSS News (2010). Available at http://dssresources.com/newsletters/264.php
27. Power, D.J.: What are some classic business DSS? DSS News (2011). Available at http://dssresources.com/faq/index.php?action=artikel&id=205
28. Power, D.J.: Decision Support, Analytics, and Business Intelligence, 2nd edn. Business Expert Press, New York, NY (2013)
29. Power, D.J.: How useful is DSS case study research? DSS News (2015). Available at http://dssresources.com/faq/index.php?action=artikel&id=345
30. Power, D.J.: Why are decision support case studies important? DSS News (2015). Available at http://dssresources.com/faq/index.php?action=artikel&id=339
31. Sauder, R.L., Westerman, W.M.: Computer aided train dispatching: decision support through optimization. Interfaces **13**(6), 24–37 (1983)
32. Schell, C.: The value of the case study as a research strategy. Working Paper Manchester Business School (1992). Available at http://www.finance-mba.com/Case%20Method.pdf
33. Scott-Morton, M.S.: Management Decision Systems: Computer-Based Support for Decision Making. Division of Research, Graduate School of Business Administration, Harvard University, Boston, MA (1971)
34. Simon, J.L.: Basic Research Methods in Social Science: The Art of Empirical Investigation. Random House, New York (1969)
35. Simons, H.: Towards a Science of the Singular: Essays About Case Study in Educational Research and Evaluation. University of East Anglia, Centre for Applied Research in Education, Norwich (1980)
36. Soy, S.K.: The case study as a research method. Unpublished Paper, University of Texas at Austin (1997). Available at https://www.ischool.utexas.edu/~ssoy/usesusers/l391d1b.htm
37. Sprague, R.H., Carlson, E.D.: Building Effective Decision Support Systems. Prentice-Hall, Englewood Cliffs, NJ (1982)
38. Stake, R.E.: The Art of Case Study Research. Sage, Thousand Oaks, CA (1995)
39. Taplin, J.M.: AAIMS: American Airlines answers the what-ifs. Infosystems, **20**(2), 40, 41, 58, 59, 68 (1973)
40. Urban, G.L.: SPRINTER: a tool for new products decision makers. Ind. Manag. Rev. **8**(2), 43–54 (1967)
41. Waterman, D.A.: A Guide to Expert Systems, pp. 237–379. Addison-Wesley, Reading, MA (1985)
42. WhatIs.com: Case Study. Project Management glossary (2015). Available at http://whatis.techtarget.com/definition/case-study
43. Wilson, S.: Explorations of the usefulness of case study evaluations. Eval. Rev. **3**(3), 446–459 (1979)
44. Yin, R.K.: Case Study Research: Design and Methods, 2nd edn. Sage, Thousand Oaks, CA (1994)
45. Yin, R.K.: Case study methods. In: Green, J.L., Camilli, G., Elmore, P.B. (eds.) Complementary Methods in Education Research, pp. 111–122. Routledge, New York (2006)
46. Yin, R.K.: Applications of Case Study Research, 3rd edn. Sage, Washington, DC (2012)

Chapter 2
ArgMed: A Support System for Medical Decision Making Based on the Analysis of Clinical Discussions

Malik Al Qassas, Daniela Fogli, Massimiliano Giacomin, and Giovanni Guida

Abstract This paper presents the design, development and experimentation of ArgMed, an interactive system aimed at supporting decision making processes that occur during clinical discussions. Clinical discussions take place on a regular basis in hospital wards and provide the forum for specialists of various medical disciplines to focus on critical cases, debate about diagnostic hypotheses, therapeutic protocols or follow-up of patient conditions, and to devise the most appropriate treatments. However, in the current medical practice, clinical discussions are usually not documented, and only the final decision is recorded on patient medical records. Therefore, some decision alternatives may get lost, the justifications for decisions made are not clarified, and the reasons in favor or against a diagnosis or a treatment remain implicit. ArgMed addresses these issues by supporting (1) the representation of discussions in a structured yet intuitive way, (2) the formalization of discussions from a logical perspective on the basis of a set of reasoning patterns (argumentation schemes) that are considered valid in the specific medical domain, (3) the identification of plausible conclusions, as well as invalid reasoning steps, hidden assumptions, or missing evidences. The paper describes the approach adopted for ArgMed design, the system architecture and operation, and the knowledge-based engine that implements decision support. The results of a preliminary experimentation of ArgMed in a real clinical environment are finally discussed.

2.1 Introduction

Clinical discussions concern the debates that physicians carry out when they need to face clinical cases that deserve specific attention and can not be dealt with by resorting to standard guidelines. The problems faced in a discussion may concern

M. Al Qassas • D. Fogli (✉) • M. Giacomin • G. Guida
University of Brescia, Brescia, Italy
e-mail: malikq2008@yahoo.com; daniela.fogli@unibs.it; massimiliano.giacomin@unibs.it; giovanni.guida@unibs.it

© Springer International Publishing Switzerland 2016

J. Papathanasiou et al. (eds.), *Real-World Decision Support Systems*,
Integrated Series in Information Systems 37,
DOI 10.1007/978-3-319-43916-7_2

diagnosis, treatment, or follow-up of patient conditions. Clinical discussions belong to a consolidated practice and generally take place once or twice a week in each specialty department of a hospital. A clinical discussion typically includes a sequence of sessions or meetings, where new cases are faced and cases already discussed in previous meetings are reconsidered, since, for example, the outcomes of new tests are available, a diagnostic hypothesis has still to be analyzed, the first effects of a treatment have to be assessed, and the treatment must be confirmed or changed.

In this paper we face the topic of clinical discussions from a practical perspective. We start from the identification of the users, we analyze their requirements, we then design and develop a system specifically tailored to meet the stated specifications, and, eventually, we experiment it in a real hospital environment.

We advocate that all the tasks physicians must face in a discussion share a common issue, namely decision making. Identifying the most likely diagnosis, selecting the best treatment, and facing possible follow-up problems timely and effectively require to take into account all information available, to apply valid reasoning patterns, and eventually to make a decision. What is needed to help physicians in these complex tasks is therefore a computer-based tool that can support their decision making processes, that is a Decision Support System (DSS).

More specifically, the DSS should support the physicians in three main tasks:

- keeping track of past discussion sessions and reviewing their main points, including the assertions made, their temporal sequence and logical relations, the decisions made, and the reasons that led to such decisions;
- establishing at each step of a discussion—or at the end of a meeting—which are the conclusions that can be logically supported by the available evidence and that might be considered as reasonable, justified decisions;
- identifying possible open problems in a discussion session, such as missing information, weak deductions not supported by enough evidence, or contradicting assertions, in order to face them in the next meeting.

Such a DSS would be not only useful, but to a large extent necessary. In fact, in the current medical practice, clinical discussions are not documented in medical records, except for the final decisions made by the medical team. As a consequence, some minor decisions may get lost as well as the justifications underlying the main decisions; the reasons in favor or against an hypothesis often remain implicit; the problems left open at the end of a meeting may be forgotten and not resumed in the next one. The impact of these issues on the quality of clinical practice should not be underestimated: it is known that inadequate medical record keeping may threaten health care quality as far as continuity of care and decision-making capabilities are concerned [23]. Moreover, the lack of documentation and logical support makes it difficult, in the unfortunate situation where a lawsuit is filed, to provide adequate justifications.

The proposed DSS, called ArgMed, is intended to be used in a hospital environment and to require only minimal training of the physicians involved. Also, it might be used by young doctors for training purposes.

In addition to facing the issue of clinical discussions from a novel perspective, ArgMed deploys new technologies. It includes three modules, namely: a Discussion Documentation Module (DDM) which is used to represent the content of a discussion session in a structured yet intuitive way, a Discussion Interpretation Module (DIM) that supports the formalization of a discussion session from a logical perspective according to a specific logical theory, and a Discussion Analysis Module (DAM), in charge of finding plausible conclusions and identifying possible logical flaws. All three modules are based on original approaches specifically designed for the clinical discussion context. DDM relies on a tree-like structure, the discussion tree, which supports a detailed representation of the assertions made during a meeting and of the temporal and logical relations among them [16]. It is strongly based on human-computer interaction methodologies [27], necessary both to support simple construction of a representation, and to allow easy understanding by all involved physicians. DIM deploys practical argumentation concepts [36], and in particular argumentation schemes [38], to represent the content of a discussion session in formal terms. Finally, DAM is based on a state-of-the-art algorithm for the computation of the justification states of a set of arguments related to a binary notion of attack [10]. As a whole, ArgMed can be classified as a knowledge-driven DSS [30], since it relies on specific domain knowledge, coded into a collection of argumentation schemes specifically designed for the medical domain, which constitutes the knowledge base of the DSS. Argumentation theory [7, 10] is adopted as the basic reasoning mechanism to process available knowledge in a decision-making framework.

ArgMed contributes to the current state of the art from several perspectives. Many Clinical Decision Support Systems (CDSSs) have been proposed to support coordination of different specialists and help them manage the huge amount of information provided by heterogeneous sources, including for example clinical guidelines and trials [34]. Other tools support the management of electronic medical records, such as WebPCR [40], LifeLines [29], and CareVis [4], and allow monitoring the state of patients under specific medical treatments. More sophisticated systems, such as REACT [19] or HT-DSS [14], help physicians perform complex planning activities. Finally, some CDSSs are able to suggest the decisions to make but without providing the relevant motivations [28], while others, such as CAPSULE [39], also provide justifications for the suggested decisions. However, the aim of all the above systems is to provide a direct support to collaboration and decision making, rather than tracking the reasons underlying the decisions of physicians. They often provide specific communication tools, such as email or chat, but they do not allow structuring actual discussions and pointing out the decisions emerging from them and the underlying motivations.

The problem of generating an electronic patient record that represents also a trace of the interactions occurred in the decision-making process is addressed in [23] through a form-based system and the use of a shared visual display. However, this system only supports collection and recording of structured data about a discussion, without providing any useful tool for interpretation and analysis.

Systems based on argumentation theory [31] have been proposed in the medical domain to help decision makers express their arguments and counterarguments, in order to solve conflicts and track the motivations underlying decisions [17]. In particular, computer-supported argumentation visualization systems aim to represent arguments in a clear and simple way, by offering users an intuitive and easy-to-use interface. These systems are usually designed for a specific application domain (such as, for example, education, law, politics) and provide different kinds of diagrammatic representations of arguments, based on graphic notations proposed in argumentation theory [2, 20, 32]. Whereas the majority of such systems are aimed at driving a discussion enforcing some constraints on the participants in the debate [2, 20, 24, 35], the focus of ArgMed is on the representation and analysis of free clinical discussions and collaborative decision making.

The paper is structured as follows. Section 2.2 describes the iterative approach followed in the development of ArgMed, while Sect. 2.3 states the main functional requirements and illustrates the overall system architecture. Section 2.4 focuses on the design and implementation of the system and includes a description of its three main modules. The user interface and the interaction between ArgMed and its users are illustrated in Sect. 2.5, while Sect. 2.6 discusses the experimentation of the system carried out at The Specialty Hospital (Amman, Jordan). Finally, Sect. 2.7 concludes the paper and provides perspectives for future work.

2.2 An Iterative Approach to System Development: From Requirements Collection to Field Testing

A user-centered design (UCD) methodology has been adopted for ArgMed development [3, 26, 27]. Attention has been focused on users' needs, characteristics, preferences and tasks, in order to develop a product able to support and improve users' current work practice and to favor system acceptability and user satisfaction [15]. Users have been involved from the very beginning in the software development life cycle through interviews, direct observation, focus groups, and prototype evaluation.

Iterative development of prototypes is the core of UCD [9]. Each prototype is evaluated with the users and, if it does not fully meet user's needs, the design process is iterated and goes through the revision of the specifications and the development of a new prototype. This iterative process is stopped only when user's needs are completely met. To implement the UCD methodology, the star life cycle of Hartson and Hix [21] has been adopted. This model includes five phases, namely: task analysis, requirements specification, design, prototyping, and implementation. The development process can start from any phase, and each phase is always followed by an evaluation activity, before proceeding to the following one. This life cycle model allowed us to gradually refine the requirements, the design, and the implemented system, on the basis of the feedback gathered from the physicians involved in clinical discussions.

More specifically, in the early phase of the project we experimented existing argumentation-based tools for discussion documentation [19, 35, 39] and we collected a first feedback from the physicians. After that, we analysed a real case study consisting of a 67-min video recording of a meeting provided by a multidisciplinary team working at the Department for Disabled People of an Italian hospital. This way, it was possible to observe and analyze in detail how the team members proposed their points of view, how the discussion evolved, and how a shared conclusion was reached in the end.

After this preliminary phase, we started the iterative development of ArgMed. Initially, it consisted in the design of paper-based and interactive prototypes, interleaved with interviews with students of the medical school and expert physicians. This activity allowed us to define a way to interactively create a discussion representation, during the discussion itself. Also, we investigated how the represented discussions could be analysed to help physicians identify weak points, missed information and possible faulty conclusions. In order to gather additional information and feedback, we examined and discussed with sample physicians some cases taken from the well-known American television medical drama “House M.D.”, which usually deals with rare pathologies and difficult diagnoses. After this phase, mostly focused on requirements analysis, the first version of ArgMed was implemented, limited to the Discussion Documentation Module. As reported in [16], an expert physician, a novice physician, and a graduate student in medicine participated in testing the first version of DDM. We then refined this module on the basis of users’ feedback and, later, the development of the Discussion Interpretation Module and of the Discussion Analysis Module started. These activities were carried out by taking into consideration several clinical discussions reported in the literature [6] as a test bench. Let us notice that the idea of organizing ArgMed in three modules emerged progressively from the iterative system development and evaluation with users. Finally, a complete version of the ArgMed prototype, developed as a web-based application, was delivered and experimented at the Specialty Hospital (Hamman, Jordan). Experimentation results are reported in Sect. 2.6.

2.3 Requirements and System Architecture

2.3.1 ArgMed Requirements

The final requirements of ArgMed, resulting at the end of the iterative process illustrated in the previous section, are summarized in Table 2.1. Requirements have been divided into four classes, one for general requirements and three for the specific modules DDM, DIM, and DAM.

Table 2.1 Requirements of ArgMed

General requirements
Manage a sequence of discussion sessions (meetings) about a given clinical case, taking place in a given period of time.
Support distributed access to all ArgMed data bases (discussion documentation, analysis, and feedback) by all users at any time.
Discussion Documentation Module
Support easy generation of discussion documentation in real time during a meeting or off-line.
Deploy a simple and intuitive visual representation for documenting a discussion (the assertions made, their authors, and their temporal and logical relations).
Generate a visual representation of a discussion understandable by all users.
Require a minimal training effort for effective use.
Discussion Interpretation Module
Support logical interpretation of a discussion after a meeting, based on the available discussion documentation and on a collection of argumentation schemes that represent usual reasoning patterns specific of the clinical domain.
Deploy a simple formal language for representing the logical structure of a discussion through a set of arguments (instantiated argumentation schemes).
Generate a visual representation of the logical structure of a discussion understandable by all users.
Require an acceptable training effort for effective use.
Allow updating the collection of the reference argumentation schemes used to interpret a discussion (delete, modify, insert new) by domain experts supported by argumentation specialists through an Argumentation Scheme Editor (ASE).
Discussion Analysis Module
Perform logical analysis of the logical structure of a discussion in order to identify conflicts, missing information, and acceptable conclusions.
Deploy a sound argumentation algorithm for performing discussion analysis.
Deploy a simple intuitive language for representing the results of discussion analysis.
Generate a representation of the results of discussion analysis understandable by all users and presented to them at the beginning of the next meeting.

More details about the meaning and implementation of such requirements are provided in the following sections.

2.3.2 System Architecture

According to the requirements stated in the previous section, the overall architecture of ArgMed comprises three main modules as illustrated in Fig. 2.1:

- The Discussion Documentation Module supports the discussion assistant, generally a young physician taking part in the meeting, in representing the content of a discussion session, namely: the assertions made, their authors, their temporal sequence, and their basic logical relations, such as conflict or support. Discussion

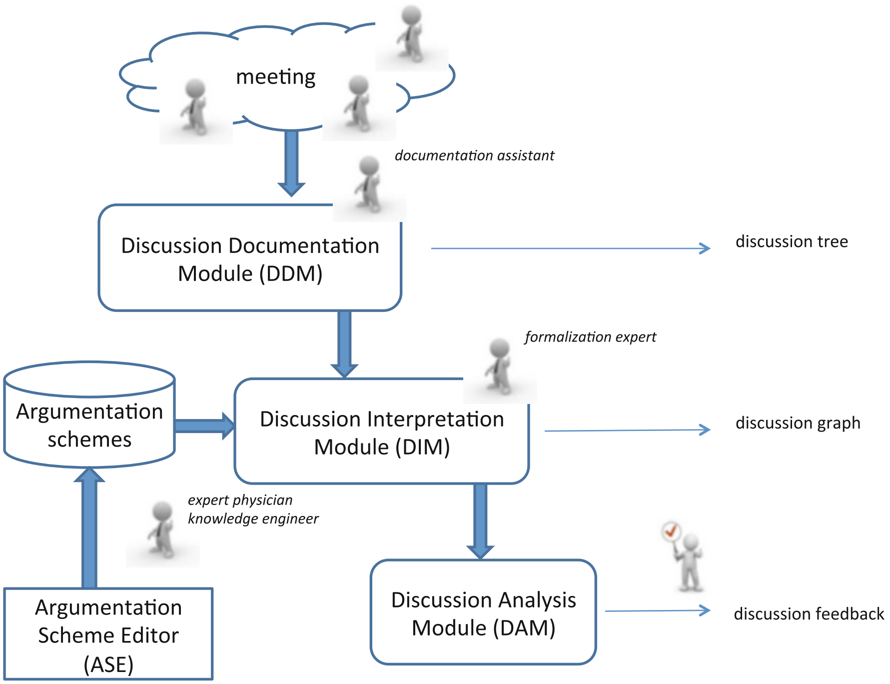


Fig. 2.1 Overall architecture of ArgMed

documentation may take place in real time during the meeting or, in case this is not possible for practical reasons, the discussion is recorded and documentation follows shortly after its conclusion.[1] The result of this activity is a *discussion tree*, that shows all relevant information about a discussion in a graphical form.

- The Discussion Interpretation Module is used by a physician specifically trained to interpret a discussion tree and formalize its content in a well defined logical formalism. Interpretation is carried out after the meeting and deploys a collection of argumentation schemes that model shared reasoning patterns specific for the medical domain. The collection of argumentation schemes that will be used for discussion interpretation is created at system design time by expert physicians supported by a knowledge engineer skilled in argumentation theory; it can be updated whenever the need to refine or drop existing argumentation schemes or to add new ones arises. The result of discussion interpretation is a *discussion graph* that shows the arguments stated during a meeting and the support and attack relations among them.

[1]From the experimentation it emerged that documenting a discussion in real-time is time consuming, thus the physicians judged the second option a more viable practice.

- The Discussion Analysis Module is in charge of processing the discussion graph in order to generate a feedback for the physicians to be taken into consideration before the next meeting. The *discussion feedback* includes: (a) plausible conclusions logically coherent with the current state of the discussion together with the relevant justifications, (b) assertions excluded and the relevant motivations, (c) possible missing information that might be useful for a deeper understanding of the issues faced. The feedback is generated automatically by a justification algorithm that produces the relevant results, presented to the physicians in an easy and natural way.

The discussion tree, discussion graph and discussion feedback of each meeting are stored in a database shared by all physicians. At any moment, and especially before the start of a new meeting, physicians can access the database, review past discussions, and focus on the feedback from the last discussion. This way, ArgMed provides physicians a focused and proactive support, not only helping them remember past discussions, but also suggesting plausible decisions and singling out possible open issues to take into consideration in the next meeting.

2.4 System Design and Implementation

2.4.1 Discussion Documentation

The Discussion Documentation Module has been designed to achieve a balance between two partially conflicting requirements: on the one hand, the module should be tailored to the physicians' habits and able to cope with a free discussion style, without constraining physicians to follow a fixed protocol or a predefined discussion scheme; on the other hand, it should support the creation of a structured representation of a discussion, in order to allow physicians to quickly recall them after some time.

The language designed for this purpose allows organizing discussion statements in a tree diagram, called discussion tree, which resembles in some way the IBIS-like notation of Rationale [33] and adopts a specific medical ontology. The types of the nodes of the discussion tree correspond to common medical concepts recurring in clinical discussions, and can be classified in two categories:

- Discussion elements, which correspond to opinions expressed by the physicians participating in the discussion. The types of discussion elements available in ArgMed are listed in Fig. 2.2 along with the icons used for their graphical representation. The name of the specialist who expressed the opinion is associated to the corresponding discussion element.
- Information elements, which represent basic information concerning the clinical case at hand and on which physicians' opinions are based (see Fig. 2.3).

ArgMED Element	Icon	Description
Clinical Case		The initial point of a clinical discussion that represents the root node of the discussion tree.
Treatment		This node corresponds to an opinion of a participant that represents a possible treatment.
Diagnosis		This node corresponds to an opinion of a participant that represents a possible diagnosis. A diagnosis is mainly based on symptoms that are usually associated with certain diseases.
Hypothesis		This node corresponds to an opinion of a participant that represents a possible non-pathological hypothesis, e.g. an explanation of symptoms that is not due to a disease but to bad habits of the patient.
Motivation PRO		This node corresponds to an opinion of a participant that represents a possible motivation to support another node.
Motivation CON		This node corresponds to an opinion of a participant that represents a possible motivation against another node.
Examination Request		This node corresponds to an opinion of a participant that a clinical examination must be performed to gather further information.

Fig. 2.2 ArgMed discussion elements and their graphical representation

ArgMED Element	Icon	Description
Personal Data		A list of all information about the patient, such as Name, Age, Sex and Address.
Symptoms		A list of all symptoms of the patient. A symptom is experienced and reported by the patient him/herself.
Semeiotics		A list of all clinical signs of the patient. A sign is discovered by the physician during patient visit.
Anamnesis		The medical history of the patient and his/her family.
Direct Observations		A list of all direct observations about the patient that physicians have seen.
Examination Reports		A list of all examination results.

Fig. 2.3 ArgMed information elements and their graphical representation

When the physician starts documenting a new discussion, the root node is generated automatically. Information elements can then be inserted by the user at any time without specific constraints, while the creation of the discussion tree is constrained by some simple composition rules that, for instance, force the user to start with a Diagnosis, a Non-pathological hypothesis or a Treatment node (for example, it is not possible to support the automatically generated node with a "Motivation PRO" node).

Any node can be connected to another node through an edge, graphically denoting that a relation holds between the two nodes. The user is not required to specify the meaning of such relation, rather the edges are simply conceived as a graphical support to recall the structure of the discussion. In particular, users are not forced to adopt terms they are not familiar with, such as "argument", "support" or "attack", even though they may implicitly express such relations during the construction of the discussion tree. For instance, linking a "Motivation PRO" node to a specific node indicates that the corresponding relation is a support and, conversely, the use of a "Motivation CON" node implies an attack relation. The feedback of physicians gathered during system development suggested that the adoption of such nodes, instead of explicitly labelling a relation as a support or an attack, was a more natural way to represent a discussion.

An example of discussion tree is reported in Sect. 2.5.2.

Of course, several concepts may remain implicit or even ambiguous in the discussion tree. One of the aims of the DIM, described in the next section, is to provide a more structured and clear representation of the discussion from a logical point of view.

2.4.2 Discussion Interpretation

The Discussion Interpretation Module supports the process of translating the discussion tree produced by the DDM into a structured logical representation that adheres to a suitable formalism. The aim is to foster discussion interpretation from a logical point of view, that is identifying the basic propositions underlying natural language statements and the relationships between them. To this purpose, we exposed physicians with several argumentation-based notations and got their feedback. On the basis of the expert opinions collected, we decided to deploy argumentation schemes [37, 38] as the basic structure of our representation language, since they turned out to be easily understandable, and physicians agreed that they support the identification of possible weak points in a discussion, such as, for example, that a possible diagnosis has been neglected, that a doubt about a diagnosis or a treatment might be raised, or that some important information to support the selection of the best treatment is missing.

An argumentation scheme is a structured representation of a reasoning pattern including, according to Walton [37], a set of premises, a conclusion, and a set of critical questions. The conclusion can generally be assumed as true if all the premises are true, however critical questions can be used to challenge the validity of the relation between the premises and the conclusion, thus providing a sieve to ensure that the reasoning pattern is applied in the correct way. In particular, when at least one critical question receives a positive answer, the relation between premises and conclusion is undermined. In our system, an argumentation scheme is a structured entity including

- the name of the argumentation scheme
- a set of parameters
- the set of premises
- the conclusion
- a set of critical questions

With respect to the structure proposed in [37], we added the set of parameters, which appear in premises, conclusions and critical questions, and which support argument creation. In other terms, assigning a value to each parameter makes it possible to instantiate the argumentation scheme to a specific case, thus yielding an argument.

The set of argumentation schemes necessary to model the specific domain of clinical discussions must be identified by a knowledge acquisition activity carried out jointly by knowledge engineers and domain experts. To support this task ArgMed includes a syntax-directed editor of argumentation schemes, namely the Argumentation Scheme Editor (ASE).

Inspired by the work of Walton [37], who introduced a comprehensive set of argumentation schemes for the legal domain, we identified a set of argumentation schemes useful to interpret clinical discussions in the medical context. To this purpose, we considered a variety of case studies, some reported in literature (for example, [11, 18, 22]) and many others found among the Case Records of the Massachusetts General Hospital [1]. In the following we report an excerpt of the argumentation schemes necessary to model discussions focusing on the identification of the best treatment for a patient. An example of the use of such argumentation schemes is reported in Sect. 2.5.3.

The Argument for Treatment Efficacy (ATE) scheme is used to model the opinion that a specific treatment would be a good choice for a patient.

ATE(P, D, T):
Parameters: P (Patient), D (disease), T (treatment).
Premise 1: Patient <P> is affected by disease <D>.
Premise 2: Treatment <T> is able to cure disease <D>.
Conclusion: Treatment <T> should be brought about for patient <P>.
CQ1: Is there an alternative treatment better than <T>?
CQ2: Is there a risk for patient <P> in following treatment <T>?
CQ3: Is there a side effect for patient <P> in following treatment <T>?

The Argument for Treatment Risk (ATR) scheme models a case where there is a risk in applying a given treatment and therefore the patient should not follow it.

ATR(P, T, N):
Parameters: P (Patient), T (treatment), N (a set of health conditions of a patient).
Premise 1: Patient <P> has conditions <N>.
Premise 2: Conditions <N> are a contraindication for treatment <T>.
Conclusion: Patient <P> should not follow treatment <T>.
CQ1: Does <P> have any specific condition that can limit the risk for <T> implied by conditions <N>?

The Argument for Risk Containment (ARC) scheme can be used to respond to the critical question of the previous ATR scheme.

ARC(P, C1, C2, T):
Parameters: P (Patient), T (treatment), C1, C2 (characteristics of a treatment).
Premise 1: Patient <P> has conditions <C2>.
Premise 2: Conditions <C2> limit the risk of treatment <T> under conditions <C1>.
Conclusion: The risk for <P> in following treatment <T> is limited.
CQ1: Is there an additional risk for patient <P> in following <T>?
CQ2: Is there a side effect for patient <P> in following <T>?

The Argument for Better Treatment (ABT) scheme models the proposal of a better treatment with respect to a previously considered one.

ABT(T1, T2, C1, C2, P, D):
Parameters: P (Patient), D (disease), T1, T2 (treatments), C1, C2 (characteristics of a treatment).
Premise 1: Patient <P> is affected by disease <D>.
Premise 2: Treatment <T1> with characteristics <C1> is able to cure disease <D>.
Premise 3: Treatment <T2> with characteristics <C2> is able to cure disease <D>.
Premise 4: Characteristics <C2> are preferable w.r.t <C1>.
Conclusion: Treatment <T2> should be brought about for patient <P>.
CQ1: Is there an alternative treatment better than <T2>?
CQ2: Is there a risk for patient <P> in following <T2>?
CQ3: Is there a side effect for patient <P> in following <T2>?

The Argument for Preference from Side Effects (APSE) scheme represents a possible reason to prefer a set of treatment characteristics over another, namely that the first set does not include some side effects that are instead included in the second set.

APSE(C1, C2, {E1, E2,. . . , En}):
Parameters: C1, C2 (characteristics of a treatment), E (side effects).
Premise 1: Characteristics <C1> include side effects <E1, E2,. . . ,En>.
Premise 2: Side effects <E1, E2. . . ,En> are not included in characteristic <C2>.
Conclusion: <C2> are preferable w.r.t <C1>.
CQ1: Are there other reasons to prefer <C1> w.r.t. <C2>?

Finally, the personal opinion of a set of physicians can be modelled by means of the following Argument from Medical Expert Opinion (AMEO) scheme.

AMEO ({PH1, PH2, PH3,. . . }, DOM, A):
Parameters: PH (physicians), DOM (medical domain), A (assertion).
Premise 1: Physicians <PH1, PH2, PH3,. . . > are specialists in domain <DOM>.
Premise 2: Physicians <PH1, PH2, PH3,. . . > assert <A>.
Conclusion: <A>.
CQ1: Is <A> inconsistent with other experts' assertions?
CQ2: Is <A> inconsistent with recent studies?
CQ3: Is there no evidence that substantiates assertion <A>?
CQ4: Is the assertion <A> not in domain <DOM>?

In order to build a logically structured representation of the reasoning activity taking place in a clinical discussion, argumentation schemes are used to interpret the assertions made by participants in a meeting. Focusing on a single fragment of a discussion at a time, the argumentation scheme that best fits the underlying reasoning path is selected and instantiated to yield an actual argument. Since each argument can be attacked or supported by others, arguments can be connected through edges that represent the relations of support or attack between them. In our approach, this can occur in two ways:

- In the instantiation phase, a relation of attack or support to other arguments can be manually defined by the user (this is the only possible way of defining support relations between arguments).

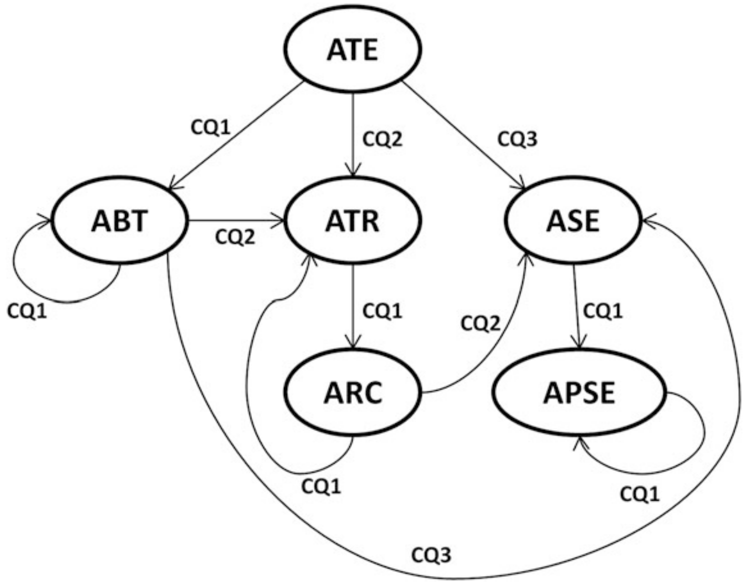


Fig. 2.4 An example of argument-relation graph for treatment-related argumentation schemes

- Attacks can be automatically derived through critical questions. More specifically, after instantiating an argumentation scheme, the user can give a positive answer to a critical question. As a consequence, another argumentation scheme is automatically instantiated by the system, generating a new argument that attacks the previous one. The argumentation scheme which is automatically instantiated is selected on the basis of an *argument-relation graph* that can be defined through the editor of argumentation schemes. Figure 2.4 shows the argument-relation graph for the argumentation schemes related to the issue of treatment selection. Each node represents an argumentation scheme and each edge represents a relation between two argumentation schemes. An edge is labeled by a critical question associated to the first argumentation scheme and whose positive answer leads to the creation of an argument that is an instance of the second argumentation scheme. The generated argument will thus attack the previous argument, namely the instance of the first argumentation scheme. In some cases, the second node is the same argumentation scheme, meaning that an argument of the same type will be instantiated (see ABT node in the figure).

As a result of the interpretation of a discussion through instantiation of argumentation schemes, a discussion graph is obtained, whose nodes represent arguments and whose edges represent support or attack relations.

2.4.3 Discussion Analysis

The aim of the Discussion Analysis Module is to process the discussion graph so as to generate a useful feedback for the user, which can support a deep understanding of the reasoning activity carried out in the discussion.

The fundamental step to generate a suitable discussion feedback is the identification of the *decision set*, namely the set of justified arguments that arise from the actual arguments posed in the discussion as well as the attack and support relations holding between them. The computation of justified arguments relies on argumentation theory, that represents a well-established logical reasoning paradigm to model common-sense discussions [8].

More specifically, the discussion graph can be translated into an *abstract argumentation framework*, namely a directed graph whose nodes correspond to arguments and whose edges represent an attack relation between them. Note that supporting arguments in the discussion graph are considered as new premises of the arguments they support; in this way, all support relations can be removed and dealt with additional attack relations. The identification of justified arguments is based on Dung's theory [13], which is widely recognized as a fundamental reference in computational argumentation for its ability to capture a variety of more specific approaches as special cases. The concept of extension plays a key role in this setting; an extension is a set of arguments that intuitively can survive the conflict together. Several definitions of extensions have been proposed in the literature, corresponding to different *argumentation semantics*: the semantics deployed in ArgMed is the *preferred semantics*, which is widely adopted in argumentation approaches [7]. In plain terms, preferred extensions correspond to the maximal sets of arguments satisfying two constraints, namely: (1) they are conflict-free, i.e. there are no attacks between arguments belonging to an extension; and (2) they are self-defending, i.e. if an argument not belonging to an extension attacks another argument belonging to the extension, it is in turn attacked by an argument of the extension itself; that is, intuitively, for any possible objection against the point of view represented by an extension, the extension is able to provide a valid response. In ArgMed, extensions are computed by the state-of-the-art solver ArgSemSat [10], which resorts to multiple calls to a Boolean Satisfiability Problem (SAT) solver in order to identify the sets satisfying the above mentioned constraints. A filtering procedure is then applied to select the maximal ones, i.e. the preferred extensions.

The Discussion Analysis Module highlights justified arguments to the user, i.e. those belonging to all preferred extensions, since their conclusions represent the decisions that can be plausibly made on the basis of the arguments present in the discussion and on their relations.

If such conclusions differ from the discussion outcome, then there is a risk that the discussion is affected by some invalid reasoning step, for example there is a counterargument to the decision made that has not been explicitly questioned.

DAM also allows the user to explore unjustified arguments as well as the relevant counterarguments, in order to have a clear view of the reasons why their conclusions have been discarded. Moreover, the user can explore the set of critical questions of justified arguments: this way, a physician might realize whether any of them did not receive sufficient attention in the discussion, thus identifying missing evidences that should be considered in the next meeting.

2.4.4 ArgMed Implementation

ArgMed has been implemented as a web-based client-server application, which ensures ubiquitous access from any site and user device, simply endowed with network connectivity and a web browser; furthermore, this architecture does not impose maintenance burdens or costs, since there is no need to install and maintain desktop applications. On the server side, a PHP web application has been developed, built on top of the Apache web server. On the client side, ArgMed has been implemented as a Rich Internet Application through the adoption of AJAX and the JQuery library. Data are stored within a MySQL DBMS and the JSON protocol is used for data exchange. In particular, argumentation schemes created through ASE and arguments generated by DIM are stored in a database compliant with the Argument Interchange Format (AIFdb) [12], developed by the scientific community to support the interchange of ideas and data between different projects and applications in the area of computational argumentation. Some extensions of AIFdb have been specifically implemented for the ArgMed project, in order to allow the instantiation of argumentation schemes and the definition of argument-relation graphs.

2.5 User Interaction

This section shows the user interface of ArgMed and describes the interaction of the user with the DDM, DIM and DAM modules. An example of a real clinical discussion is first introduced in Sect. 2.5.1 to be used as running example.

2.5.1 A Clinical Discussion

The discussion considered here as a running example concerns a case of appendicular abscess; it has been provided by the physicians participating in the experimentation of ArgMed at The Speciality Hospital (see Sect. 2.6). The case is reported in Table 2.2. It involves a team of surgeons, namely a chief surgeon (S1) and some seniors, juniors, and students in general surgery (S2,..,S5), who discuss about finding the best treatment to apply to a young patient with an appendicular abscess. The discussion starts with the presentation of the case and develops with each

Table 2.2 The appendicular abscess case

Presentation of the case
A 7 year old male complaining of abdominal pain and fever for the past 5 days. He is also complaining of recurrent vomiting. The abdominal pain is diffuse lower abdominal pain and is associated with diarrhea. Patient report and physical examination is compatible with complicated appendicitis. Ultrasound and CT-scan show an appendicular inflammatory mass and small fluid collection 3*3 cm in the right iliac fossa. WBC (white blood cell count): 17,000 and temperature: 38.9. Vital signs are within normal range (Blood pressure, Heart rate, respiratory rate).
First meeting
S1,A1: My opinion is to drain the fluid collection and treat with IV antibiotics. This has a good cure rate and the patient will remain stable.
S2,A2: I agree with S1, this treatment has a good cure rate.
S3,A3: From our past experience, we have always performed this treatment, so I agree with S1 and my preference is to do the same.
S4,A4: It's only been 5 days since symptoms started, I believe the mass is not well formed (from ultrasound and CT-scan), and the appendix can be easily excised and we'll shorten the patients hospital stay.
S5,A5: I agree with S4, the appendix can easily be excised and we'll shorten the patients hospital stay.
S1,A6: An operation would be risky with an appendicular mass. But you don't need to do an interval appendectomy and the fluid collection is too small to be drained; we can cover with antibiotics and wait.
S4,A7: I believe that we have to repeat the ultrasound and CT-scan to better analyze the size of inflammatory mass and the fluid collection, and then decide the most appropriate treatment for this case.
S1,A8: I agree with S4.
S5,A9: I agree with S4, It's better if we repeat the examinations.
Second meeting
S1,A10: From the ultrasound and the CT-scan we have the same result (a small appendicular inflammatory mass and a small fluid collection 3*3 cm in the right iliac fossa), so my opinion is to treat with IV antibiotics without the drainage of the fluid collection.
S2,A11: I agree with S1, an operation would be risky to a child patient with an appendicular mass.
S5,A12: I agree with S1, there are new medical studies reporting that if we treat with IV antibiotics, we will have a good cure rate.
Final decision
Treat the patient with IV antibiotics if he remains stable. Consult the patient that 10 % will recur as simple acute appendicitis and they can be treated accordingly.

participant proposing his/her opinion about how to cure the patient. In the first session of the discussion, the surgeons decide to repeat some examinations (see statement A9 in Table 2.2) and in the next session they decide the best treatment for the patient (antibiotics). The next sub-sections illustrate ArgMed user interface and operation with the help of a portion of this discussion.

2.5.2 Discussion Documentation

Figure 2.5 shows a screenshot of the Discussion Documentation Module. The central area is organized according to a tab-based structure: in the first tab, called “Discussion”, the user can compose the discussion tree. To this end, the user can drag-and-drop the discussion elements available in the left area, thus creating a diagnosis, a treatment, a non-pathological hypothesis, an examination request, or a motivation in favour (pro) or against (con), i.e. an explicit assertion that corroborates or contradicts a given statement. The user can also access information elements (Personal Data, Symptoms, Semeiotics, Anamnesis, Direct Observations, Examination Reports) available in the other tabs and select them to be included in the discussion tree as nodes in favor or against other nodes of the discussion.

In particular, in Fig. 2.5 the Discussion tab includes a representation of the first four statements of the sample case study. A treatment node corresponding to statement A1 “My opinion is to drain the fluid collection and treat with IV antibiotics. This has a good cure rate and the patient will remain stable” has been created and connected to the root of the discussion tree. Then, statements A2 and A3, which provide motivations for supporting statement A1, have been associated with two Motivation PRO nodes connected to the treatment node. Statement A4 “It’s only been 5 days since symptoms started, I believe the mass is not well formed (from ultrasound and CT-scan), and the appendix can be easily excised and we’ll

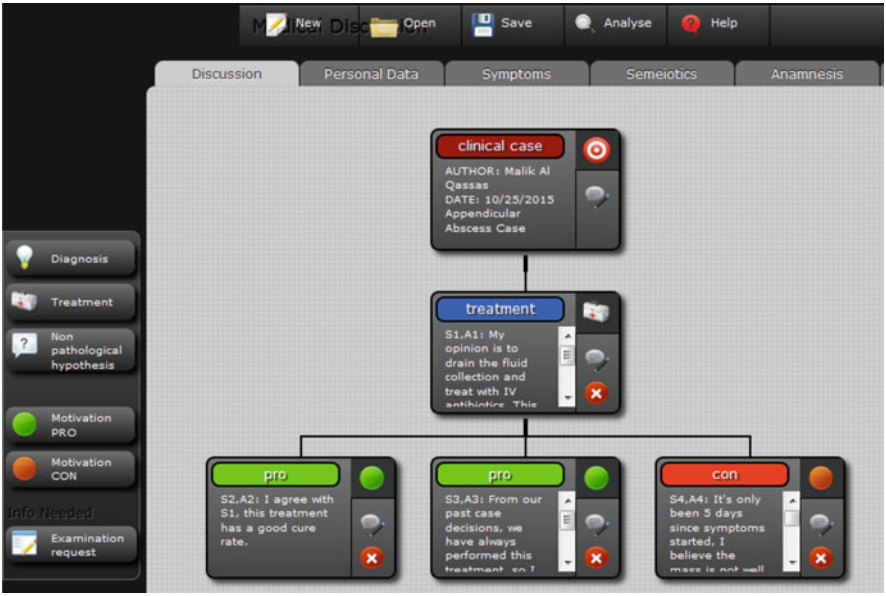


Fig. 2.5 The discussion tree representing the first four statements of the appendicular abscess case

shorten the patients hospital stay" proposes an alternative treatment with respect to that indicated in statement A1. Therefore, since statement A4 actually attacks A1, a new Motivation CON node has been created and connected to the node A1.

2.5.3 Discussion Interpretation

After a discussion has been carefully documented, the user can formalize its content through the Documentation Interpretation Module. Figure 2.6 shows a screenshot of this module, where the discussion statements are visualized as a multi-level list in the left part of the screen. The user may select these statements and create a set of arguments related to the discussion by instantiating suitable argumentation schemes.

In Fig. 2.6, the user is creating a new argument associated with statement A1. On user request to create a new argument, the system asks him/her the argument name and the type of argument (that is the argumentation scheme that must be instantiated). Moreover, if the argument has to be connected to another argument, the user can indicate the later argument and the relation type (attack, support or neutral) of the connection. For statement A1, the user has inserted "Arg1" as argument name and has selected ATE ("Argument from Treatment Efficacy") among the available argumentation schemes and "Neutral" as relation type, since it is the first argument created for this discussion. The user can edit the argument parameters by clicking on the words in bold to activate a pop-up allowing parameter value setting (see Fig. 2.6). Figure 2.7 shows the elements of argument Arg1 with the parameters edited by the user. Here, the user has assigned the parameter "Disease" the value "appendicular abscess" and the parameter "Treatment" the value "drainage and antibiotics".

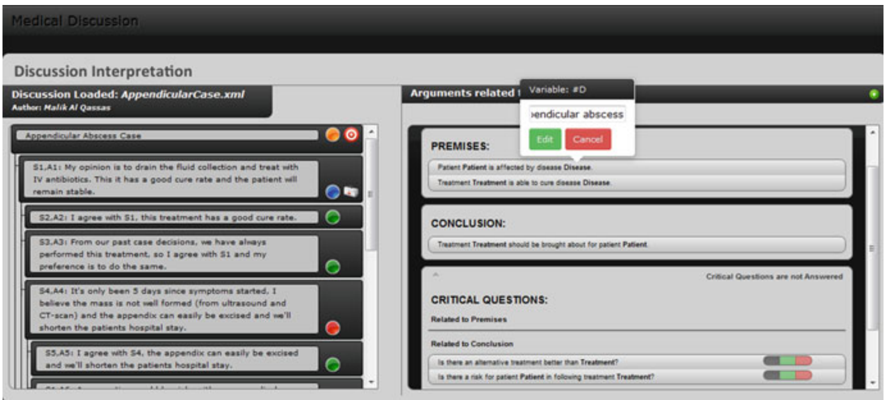


Fig. 2.6 The user is editing one parameter of "Arg1" created for the statement A1

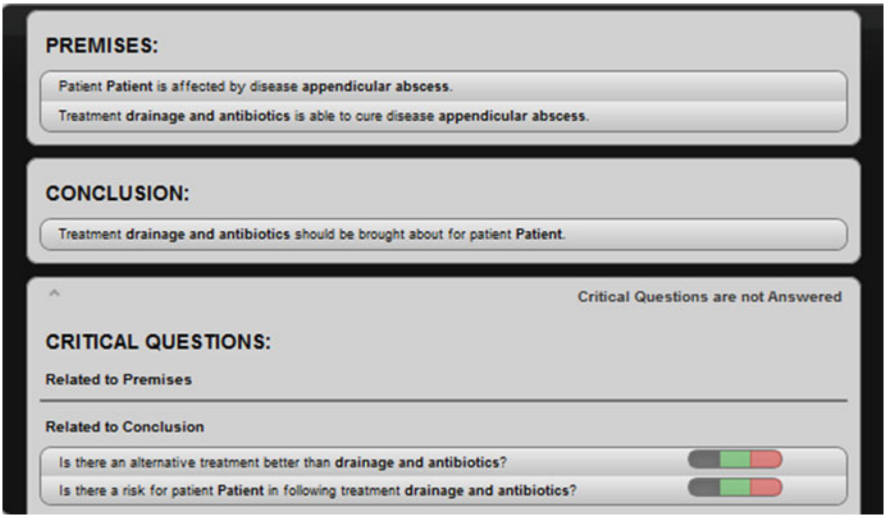


Fig. 2.7 Arg1 as a instantiation of argumentation scheme "Argument from Treatment Efficacy"

The argumentation scheme AMEO ("Argument from Medical Expert Opinion") can be used to represent statements A2 "I agree with S1, this treatment has a good cure rate" and A3 "From our past experience, we have always performed this treatment, so I agree with S1 and my preference is to do the same". This way argument Arg2 is created and a relation of "Pro" type with argument Arg1 is defined. For this argument, the user may instantiate the parameters as follows: "Physicians"="{S2, S3}", "Domain"="surgery", "Assertion"="drainage and antibiotics are able to cure the appendicular abscess".

Statement A4 "It's only been 5 days since symptoms started, I believe the mass is not well formed (from ultrasound and CT-scan) and the appendix can easily be excised and we'll shorten the patients hospital stay" provides an answer to critical question CQ1 "Is there an alternative treatment better than drainage and antibiotics?" of Arg1. When CQ1 of Arg1 receives a positive answer, the system, on the basis of the argument-relation graph (Fig. 2.4), instantiates the argumentation scheme ABT ("Argument for Better Treatment"), giving rise to a new argument (Arg3), which attacks argument Arg1.

The other statements of the discussion can be interpreted by the user in a similar way, by eventually generating the discussion graph.

2.5.4 Discussion Analysis

Figure 2.8 shows a screenshot of the Discussion Analysis Module. In the left area the discussion graph for the appendicular abscess case is shown. The graph has

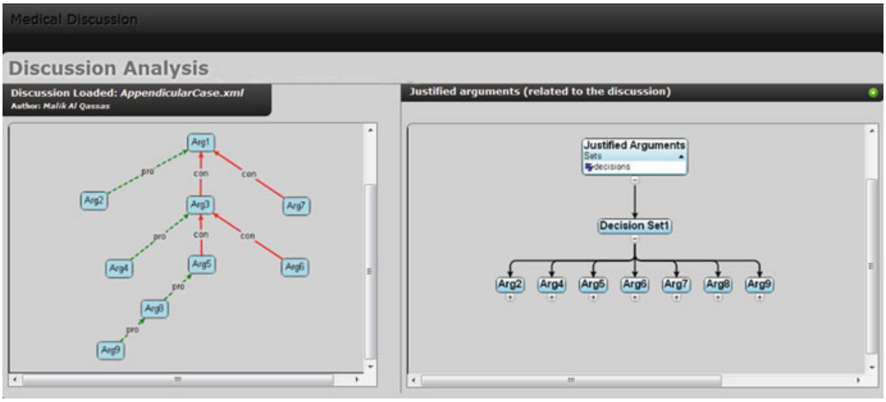


Fig. 2.8 The analysis result of the Appendicular Abscess case: discussion graph (*left*) and decision set (*right*)

then been processed through the ArgSemSat solver, and the decision set has been identified including the arguments that can be considered as justified (see Fig. 2.8, right area). Here, arguments Arg2, Arg4, Arg5, Arg6, Arg7, Arg8 and Arg9 are justified.

By clicking on the "+" button under each argument name, the user can view the details of the argument. The user can also filter the decisions by configuration type: for example, by selecting "treatment", only the conclusions related to treatments are shown. In this case, according to the relevant conclusions (in particular, those of Arg5 and Arg7), "antibiotics" should be brought about among the considered treatments. This corresponds to the outcome of the discussion, confirming that all possible counterarguments explicitly advanced in the discussion have been properly considered.

Of course, analysing the critical questions and how they have been tackled in the discussion, the user may always realize that alternative options did not receive proper consideration in the discussion, thus they should be examined in the next meeting.

2.6 ArgMed Experimentation

Experimentation of ArgMed was carried in July-August 2015 at the Specialty Hospital, a leading private teaching hospital, located in Amman (Jordan).

Experimentation consisted of four activities:

- indirect functional assessment through observation of multi-disciplinary meetings;
- direct user tests, carried out by selected physicians;

- structured after-test interviews;
- elaboration of new case studies provided by selected physicians.

Before carrying out direct user tests, the overall adequacy of ArgMed to support documentation, interpretation and analysis of clinical discussions was verified through a non-intrusive observation technique. To this purpose, one of the authors of this paper participated in 15 multi-disciplinary meetings to observe physicians' work and assess strong and weak points of ArgMed. At Specialty Hospital, physicians regularly meet every morning (5 times/week) and discuss 3 to 5 cases in each meeting. The participants in this meeting are a chief physician (more than 15 years of experience), senior physicians (3–5 years of experience), junior physicians (1–3 years of experience), interns and students. The observation activity allowed us to get a deeper understanding of the structure of clinical discussions, of the way they developed, and of the process that led to final decision making. As a result, a first validation of the main functions offered by ArgMed and of its overall mode of operation was obtained.

Then, a senior surgeon, a junior surgeon and a junior internist participated in direct test sessions with ArgMed. After a brief presentation of the tool, physicians were trained in the use of ArgMed through the analysis of simple cases. A literature case study [11] about the treatment of larynx cancer was presented to them to be used as a test case. After that, two test sessions took place. In the first session users were asked to carry out specific tasks, such as representing a few assertions through the documentation language, interpreting assertions through available argumentation schemes, and launching the analysis step. In the second session, users were required to autonomously create, document and analyze a substantial fragment of the discussion about the treatment of larynx cancer.

In both test sessions, a thinking-aloud observation protocol was adopted to collect as much as possible feedback from the users, including reasoning strategies, the way of interacting with the system, the problems met, and any kind of comments and suggestions.

In the first test session physicians showed a positive attitude towards the system and a critical approach to task execution. The observer clarified that there were several ways to model a discussion and that each physician could make his/her choices, obviously without changing the discussion logic. The users appreciated the drag-and-drop interaction with the Discussion Documentation Module that allowed them to create the discussion tree in an easy and natural way. All of them pointed out instead the need for further training in the use of argumentation schemes, which turned out to be rather demanding.

The second test session, held after administration of a supplementary training, has been useful to investigate a more realistic use of the system. Physicians were able to easily create both the discussion tree and the discussion graph and to carry out discussion analysis to obtain a decision suggestion. The decision proposed by the system was considered plausible and useful.

After user tests, a structured interview was carried out with each participant to gather a more formal feedback about the usefulness and usability of ArgMed. The interview consisted of two groups of questions:

1. Usefulness
   (a) Who is the ideal user of ArgMed (chief, seniors, juniors, interns, students)?
   (b) How could ArgMed be integrated in the current work practice? (real-time, off-line)
   (c) Do you think that critical questions are a useful tool for focusing attention on important issues and keeping the discussion on a sound logical track?
   (d) Which advantages do you see in the adoption of ArgMed for discussion documentation?
   (e) Which advantages do you see in the adoption of ArgMed for discussion analysis?
   (f) Which drawbacks do you see in the adoption of ArgMed for discussion documentation?
   (g) Which drawbacks do you see in the adoption of ArgMed for discussion analysis?
   (h) Have you suggestions about additional or alternative argumentation schemes?
2. Usability
   (a) Do you think discussion visualization as a discussion tree is clear and easy to understand for all participants in a clinical discussion?
   (b) Is the discussion graph suitable for identifying the global logical structure of a discussion and for justifying the suggested conclusion?
   (c) Do you think that ArgMed is easy to learn?
   (d) Do you judge ArgMed as an efficient or time-consuming tool?
   (e) Have you practical suggestions about how to improve the tool?

Table 2.3 shows the results of these structured interviews. As one can notice, participants provided very positive assessments, yet suggesting several improvements. Important remarks were made about the integration of ArgMed with the current clinical practice: in particular, participants suggested that the tool could be used after the meetings (off-line), to remember discussion details and record analyses of several discussions for future reference. ArgMed was judged very useful for consultation physicians (namely, specialists who cannot participate in the meetings), who could review the discussion off-line and insert their opinions. All interviewees highlighted the high potential of the system as an educational tool, to train students and junior physicians in clinical case analysis.

All interviewees agreed that the discussion tree was a good representation of clinical discussions, even though participants highlighted that creating such tree may be time-consuming, especially if the system is used for real-time documentation. The discussion graph was also considered very useful for identifying possible missing information, for analyzing contrasting points of view, and for focusing the final decision of a discussion.

Table 2.3 The results of the structured interviews

	Senior surgeon	Junior surgeon	Junior internist
1.a	Juniors followed by seniors	Students in training followed by seniors	Interns and students
1.b	Off-line	Off-line	Off-line
1.c	Yes	Yes	Yes
1.d	Useful as reference for future discussions and similar cases, reduce paper usage and can be used as educational tool	Useful as reference for future cases and for training purposes	Useful for teaching purposes and as patient records for future references
1.e	Useful for training purposes	Useful as reference for future discussion analyses, useful for consultation requests, useful for training	Useful to take better decisions; useful in particular for interns and students in training
1.f	It is time-consuming if used at real-time	Creating the discussion tree is time-consuming	The discussion should be documented after the meeting is closed, thus some paper notes or video recording are needed
1.g	It does not consider the role levels of users	It does not manage the credibility of users	It does not manage the mistakes of users
1.h	Argument from history, Argument from consultation, Argument from physical examinations, Argument from laboratory examination	Argument from complications, Argument from history, Argument from vital signs, Argument from consultation	Argument from consultation
2.a	Yes, it is very effective	Yes	Yes
2.b	Yes, it is useful for analyzing contrasting opinions	Yes, but interaction with nodes must be made more intuitive	Yes, it allows focusing the final decision
2.c	Discussion documentation is easy to learn, but discussion interpretation and analysis require an expert in argumentation	Analysis requires some training in argumentation theory	Discussion documentation is easy to learn, interpretation through argumentation schemes needs several training sessions
2.d	Time-consuming	Time-consuming	Time-consuming
2.e	Add a search engine for clinical case retrieval, create a tablet-oriented version	Integrate protocols and guidelines, create a large database of discussions	Add option to print the discussion with suggested decisions as a PDF file

Participants said they found the argumentation schemes easy to learn and adequate to clinical discussion analysis. They also proposed some new argumentation schemes to be added to the initial set available in ArgMed, useful for a more detailed logical interpretation of clinical discussions. Furthermore, they noted that ArgMed did not take into consideration the different roles and experience of participants in a discussion, as well as their credibility. They then suggested to include in ArgMed standard clinical guidelines and local medical protocols in order to take them into account during a discussion and to assess the compliance of the final decisions made. Finally, some practical suggestions were proposed to improve the tool, such as: creating a tablet-oriented version, adding an option to print a discussion and all related materials as a PDF file, creating a structured database of the discussions held, designing a search engine for clinical case retrieval.

In the fourth experimentation activity, the physicians provided us with three further case studies involving discussions they encountered in recent meetings they attended. We elaborated with them such discussions in order to evaluate once again whether ArgMed was suitable and useful for discussion documentation and interpretation, as well as a support tool for decision making. Even though we discovered that some additional argumentation schemes needed to be included in the system, the functions available in ArgMed allowed managing successfully all the three case studies. As a consequence, we chose one of them, the appendicular abscess case, as running example for the present paper.

2.7 Conclusions

In this chapter we have presented a CDSS aimed at supporting a consolidated practice of hospitals and other healthcare facilities, that is, multidisciplinary team meetings that occur regularly once or twice per week to discuss the most serious clinical cases [22]. To our knowledge, CDSSs with a similar goal have not been proposed yet, even though their importance has been recently underlined in literature [25]. Indeed, as far as multidisciplinary team meetings are concerned, literature works address the problem from a different perspective: they are mainly aimed at facilitating the interaction among physicians through the use of advanced multimedia technologies (see for instance [18]), rather than supporting their decision making tasks.

Furthermore, ArgMed supports clinical decision making in an original way with respect to traditional CDSSs. The latter are usually based on knowledge about diagnoses or treatments for specific diseases or about protocols to be followed in specific situations. ArgMed, instead, is based on higher-level knowledge about physicians' reasoning patterns. We collected and distilled this knowledge from observations of real clinical discussions, interviews with physicians, and analysis of several clinical cases reported in literature. Such knowledge, represented as a set of argumentation schemes, is used in ArgMed for discussion interpretation and analysis, in order to provide suggestions about the decision to make or to shed light on weak points in reasoning paths. More precisely, at the end of discussion analysis, the system is able to provide physicians with different kinds of support: (a) suggestions for decision making, in the case only one argument related to a specific diagnosis or treatment is justified; (b) warnings about possible alternative conclusions, whenever several arguments associated to alternative diagnoses or treatments are equally justified; (c) suggestions for considering the critical questions that have remained unanswered, in order to make hidden assumptions explicit or to gather additional information about the case under discussion.

From a more general perspective, ArgMed may plays three different functions in real clinical settings [5]. First, it may play the function of "memory", since it allows creating orderly records of all the statements made by physicians, as well as keeping trace of their temporal and logical articulation. Physicians can hardly remind the details of previous discussions and can greatly benefit from the availability of a system that helps avoid returning again on issues that have been already solved in past meetings, and prevent incurring into inconsistencies or contradictions. Second, it may sustain "reflection", because once a discussion has been represented as a discussion tree, it can be interpreted and analysed to perform automated reasoning activities, including the identification of possible contradictions and the discovery of invalid reasoning paths (missing reasoning steps, hidden assumptions, missing evidences, etc.). The results of this analysis can support a more correct and logically sound evolution of their reasoning in the following working sessions. Third, a "generalization" activity can be performed with ArgMed: indeed, at the end of the decision making process, it is possible to reconsider the whole series of discussions held, the decisions made and their practical effects, in order to derive a best practice of general validity, which might be adopted in similar cases. This important function may not only help physicians define improved decision procedures, but may also constitute the core function of an education tool for clinical case analysis, which can be used for a more effective and focused training of young physicians.

Promising directions for future research include four main issues:

- the design of more effective and user-oriented tools to present discussion feedback; this is indeed an important point to support the practical use and the effectiveness of ArgMed;
- the extension, refinement and validation of the collection of argumentation schemes available for discussion interpretation; the elicitation of the common reasoning paths used by physicians is not an easy task and the quality of the argumentation schemes has a strong impact both on the interpretation of a discussion and on the results of the analysis;
- the comparison of the decisions made in a discussion with the relevant clinical guideline, either to correct the conclusions reached or to improve the guideline;
- the discovery from the argumentation process of new clinical knowledge to be reused in similar cases that might arise in the future.

References

1. Rosenberg, E.S., Harris, N.L. (eds.): Case records of the Massachusetts general hospital. N. Engl. J. Med. (2016). doi:10.1056/ NEJMcpc1415172
2. Compendium web site. http://compendium.open.ac.uk/index.html (2012)
3. Abras, C., Maloney-Krichmar, D., Preece, J.: User-Centered Design. Sage, Thousand Oaks (2004)
4. Aigner, W., Miksch, S.: CareVis: integrated visualization of computerized protocols and temporal patient data. Artif. Intell. Med. **37**(3), 203–218 (2006)
5. Al Qassas, M., Fogli, D., Giacomin, M., Guida, G.: Supporting medical decision making through the analysis of clinical discussions. In: Phillips-Wren, G., Carlsson, S., Respicio, A., Brezillon, P. (eds.) DSS 2.0 - Supporting Decision Making with New Technologies, pp. 42–53. IOS Press, Paris (2014)
6. Al Qassas, M., Fogli, D., Giacomin, M., Guida, G.: Analysis of clinical discussions based on argumentation schemes. Procedia Comput. Sci. **64**, 282–289 (2015)
7. Baroni, P., Caminada, M., Giacomin, M.: An introduction to argumentation semantics. Knowl. Eng. Rev. **26**(4), 365–410 (2011)
8. Bench-Capon, T., Dunne, P.E.: Argumentation in artificial intelligence. Artif. Intell. **171**(10–15), 619–641 (2007)
9. Bødker, S., Grønbæk, K.: Cooperative prototyping: users and designers in mutual activity, pp. 331–358. Academic, London (1991)
10. Cerutti, F., Dunne, P.E., Giacomin, M., Vallati, M.: Computing preferred extensions in abstract argumentation: a SAT-based approach. In: TAFA 2013. Lecture Notes in Computer Science, vol. 8306, pp. 176–193. Springer, Berlin, Heidelberg (2014)
11. Chang, C.F., Miller, A., Ghose, A.: Mixed-initiative argumentation: group decision support in medicine. In: Electronic Healthcare - Second International ICST Conference, eHealth 2009, Istanbul, Turkey, 23–25 September, Revised Selected Papers, pp. 43–50 (2009)
12. Chesñevar, C., McGinnis, J., Modgil, S., Rahwan, I., Reed, C., Simari, G., South, M., Vreeswijk, G., Willmott, S.: Towards an argument interchange format. Knowl. Eng. Rev. **21**(4), 293–316 (2006)
13. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and n-Person games. Artif. Intell. **77**(2), 321–357 (1995)
14. Fogli, D., Guida, G.: Enabling collaboration in emergency management through a knowledge-based decision support system. In: Respicio, A., Burstein, F. (eds.) Fusing Decision Support Systems into the Fabric of the Context, pp. 291–302. IOS Press, Amsterdam (2012)
15. Fogli, D., Guida, G.: Knowledge-centered design of decision support systems for emergency management. Decis. Support Syst. **55**, 336–347 (2013)
16. Fogli, D., Giacomin, M., Stocco, F., Vivenzi, F.: Supporting medical discussions through an argumentation-based tool. In: Proceedings of the Biannual Conference of the Italian Chapter of SIGCHI (CHItaly2013), pp. 18:1–18:10. ACM Press, New York (2013)
17. Fox, J., Glasspool, D.W., Grecu, D., Modgil, S., South, M., Patkar, V.: Argumentation-based inference and decision making - a medical perspective. IEEE Intell. Syst. **6**(22), 34–41 (2007)
18. Frykholm, O., Groth, K.: References to personal experiences and scientific evidence during medical multi-disciplinary team meetings. Behav. Inform. Technol. **30**(4), 455–466 (2011)
19. Glasspool, D.W., Oettinger, A., Smith-Spark, J.H., Castillo, F.D., Monaghan, V.E.L., Fox, J.: Supporting medical planning by mitigating cognitive load. Methods Inf. Med. **46**(6), 636–640 (2007)
20. Gordon, T.F.: An overview of the carneades argumentation support system. In: Reed, C., Tindale, C.W. (eds.) Dialectis, Dialogue and Argumentation: An Examination of Douglas Walton's Theories of Reasoning, pp. 145–156. College Publications, London (2010)
21. Hartson, H.R., Hix, D.: Human-computer interface development: concepts and systems for its management. ACM Comput. Surv. **21**(1), 5–92 (1989)

22. Kane, B., Luz, S.: Achieving diagnosis by consensus. Comput. Supported Coop. Work **18**(4), 357–392 (2009)
23. Kane, B.T., Toussaint, P.J., Luz, S.: Shared decision making needs a communication record. In: Proceedings of the 16th ACM Conference on Computer Supported Cooperative Work (CSCW '13), pp. 79–90. ACM Press, New York (2013)
24. Karamanou, A., Loutas, N., Tarabanis, K.A.: ArgVis: structuring political deliberations using innovative visualization technologies. In: Tambouris, E., Macintosh, A., de Bruijn, H. (eds.) Proceedings of Electronic Participation, Third IFIP WG 8.5 Int. Conf. ePart 2001. Lecture Notes in Computer Science, vol. 6847, pp. 87–98. Springer, Delft (2011)
25. Luz, S., Kane, B.: Perspectives on intelligent systems support for multidisciplinary medical teams. In: AAAI Spring Symposium Series, pp. 272–275. Springer, Berlin (2016)
26. Moran, T.P.: Design Rationale: Concepts, Techniques, and Use. L. Erlbaum Associates Inc., Hillsdale, NJ (1996)
27. Norman, D.A., Draper, S.W.: User-Centered System Design: New perspectives on Human-Computer Interaction. Lawrence Erlbaum Associates, Hillsdale, NJ (1986)
28. Peleg, M., Wang, D., Fodor, A., Keren, S., Karnieli, E.: Adaptation of practice guidelines for clinical decision support: a case study of diabetic foot care. In: Proceedings Biennal European Conference on Artificial Intelligence (ECAI). Riva del Garda, Italy (2006)
29. Plaisant, C., Mushlin, R., Snyder, A., Li, J., Heller, D., Shneiderman, B.: LifeLines: using visualization to enhance navigation and analysis of patient records. In: Proceedings American Medical Informatic Association Annual Fall Symposium, Orlando, USA, pp. 76–80 (1998)
30. Power, D.J.: Computerized decision support case study research: concepts and suggestions. In: Real-World Decision Support Systems Cases Studies. Information Management. Springer, Berlin (2016)
31. Prakken, H., Vreeswijk, G.: Logics for defeasible argumentation. In: Gabbay, D., Guenthner, F. (eds.) Handbook of Philosophical Logic, pp. 219–318. Kluwer Academic, Dordrecht (2002)
32. Reed, C.A., Rowe, G.W.A.: Araucaria: software for argument analysis, diagramming and representation. Int. J. AI Tools **13**(4), 961–980 (2004)
33. Sbarski, P., van Gelder, T., Marriott, K., Prager, D., Bulka, A.: Visualizing argument structure. In: Bebis, G. et al. (ed.) Proceedings of the Fourth Int'l Symposium Advances in Visual Computing (ISCV '08). Lecture Notes in Computer Science, vol. 5358, pp. 129–138 (2008)
34. Sullivan, F., Wyatt, J.C.: How decision support tools help define clinical problems. Br. Med. J. **331**(7520), 831–833 (2005)
35. Tolchinsky, P., Cortés, U., Modgil, S., Caballero, F., López-Navidad, A.: Increasing human-organ transplant availability: argumentation-based agent deliberation. IEEE Intell. Syst. **21**(6), 30–37 (2006)
36. Van Eemeren, F.H., Grootendorst, R., Snoeck Henkemans, F.: Fundamentals of Argumentation Theory: A Handbook of Historical Background and Contemporary Developments. Lawrence Erlbaum Associates, Mahwah, NJ (1996)
37. Walton, D.: Argumentation Schemes for Presumptive Reasoning. Lawrence Erlbaum Associates, Mahwah, NJ (2006)
38. Walton, D., Reed, C., Macagno, F.: Argumentation Schemes. Cambridge University Press, Cambridge (2008)
39. Walton, R., Gierl, C., Yudkin, P., Mistry, H., Vessey, M.P., Fox, J.: Evaluation of computer support for prescribing (CAPSULE) using simulated cases. Br. Med. J. **315**, 791–795 (1997)
40. Web browser based patient care report. http://emsfiresoftware.com/products/wpcr/ (2004)

Chapter 3
The Integration of Decision Analysis Techniques in High-Throughput Clinical Analyzers

Anna Arigliano, Pierpaolo Caricato, Antonio Grieco, and Emanuela Guerriero

Abstract From the early 1990s, the introduction of high-throughput clinical analyzers has significantly changed the workflow of In-Vitro-Diagnostics (IVD) tests. These high-tech instruments have helped and keep helping clinical laboratories both to increase quality diagnostic responses and to get more for every dollar they spend. Nevertheless, IVD industrial research has been up to now largely *hardware-driven* with the introduction in the market of many sophisticated technologies. The *software component*, models and decision support systems in particular, has lagged behind. To reach the full potential of diagnostic automation, it must be addressed the challenge of making the most intelligent use of the hardware that is deployed. Focusing on time efficiency, the authors have devised an operations research-based method for a class of high-throughput clinical analyzers. To demonstrate the validity of the research, the proposed method has been coded and integrated into the Laboratory Information System of the *Laboratorio di Analisi Cliniche Dr. P. Pignatelli*, one of the most important clinical laboratories in Southern Italy. Siemens Immulite®2000 has been the reference case. The enhanced operating planning procedure provides a monetary benefit of 52,000 USD/year per instruments and a trade-off between clinical benefits and operating costs equivalent to the one provided by the current hardware-driven research at Siemens. Despite the proposed approach has the potential to determine guidelines for enhancing a wide range of current high-throughput clinical analyzers, we have to register a failure in trying to convince technology providers to invest in embedding such new models in their hardware. Some possible causes for such failure are highlighted, trying to find possible improvements for future developments.

A. Arigliano • P. Caricato (✉) • A. Grieco • E. Guerriero
DII – Università del Salento, Via Monteroni, Lecce, Italy
e-mail: pierpaolo.caricato@unisalento.it

© Springer International Publishing Switzerland 2016

J. Papathanasiou et al. (eds.), *Real-World Decision Support Systems*,
Integrated Series in Information Systems 37,
DOI 10.1007/978-3-319-43916-7_3

3.1 Introduction

Nowadays healthcare providers worldwide increasingly recognize the power of instrument-based diagnostics to enhance the efficiency and the effectiveness of healthcare management. Instrument-based diagnostic phase is almost exclusively executed by imaging techniques and/or by In-Vitro-Diagnostics (IVD) tests. IVD tests are carried out in clinical laboratories on blood or other bodily fluids and tissue samples.

Focusing on IVD tests and on their impact on disease management, a central role is played by tests derived from immunology and molecular biology, known as immunoassays. Immunoassays provide precise, rapid identification and measurement of chemical substances (i.e. analytes) relevant from the medical point of view. Hormones, cytokines, vitamins, drugs, transport proteins, antibodies and biochemical markers of viruses or microorganisms are the main analytes detected by immunoassays.

In the last 10 years, several important healthcare breakthroughs have had an important impact on immunoassays. The advances in the human genome project and functional genomics have allowed to define new and better markers for the most pressing diseases like cancer and diabetes. These advances contributes to disease treatment efficiency, to increase the number of new immunoassays as well as their demand. In USA alone, immunoassays are available for more than 2000 diseases. At the same time, immunoassays are the fastest growing IVD segment with a compound annual growth rate of 16 %.

The continuous growth of the IVD immunoassay industry reflects the rising importance of the corresponding automation technologies. During the last years, laboratory automation solutions, based on high-throughput immunoassay analyzers, have been widely deployed in laboratories workflow. High-throughput analyzers are electro-mechanical devices able to automatically execute a set of analytical procedures, i.e. the set of processing steps required to determine and measure a specific analyte. A clinical analyzer is classified as high-throughput if the number of started assays per hour is of the order of hundreds.

A clinical laboratory typically processes analysis requirements with different priority levels, based on the patient's severity. The samples are typically collected during the early part of the morning, while the actual processing of the samples takes place during the rest of the day, using high-throughput clinical analyzers.

The first decision problem to be addressed is to determine the workload of the available analyzers during the day. This can be modeled as the classical loading problem that often arises in manufacturing production management: the analyzers in the laboratory can be seen as the machines available in a shop floor, while the required clinical tests represent the tasks to be processed. The number of jobs that can be simultaneously loaded on a machine typically exceed its capacity, hence a selection must be done among the required tests, to decide which ones will be processed first.

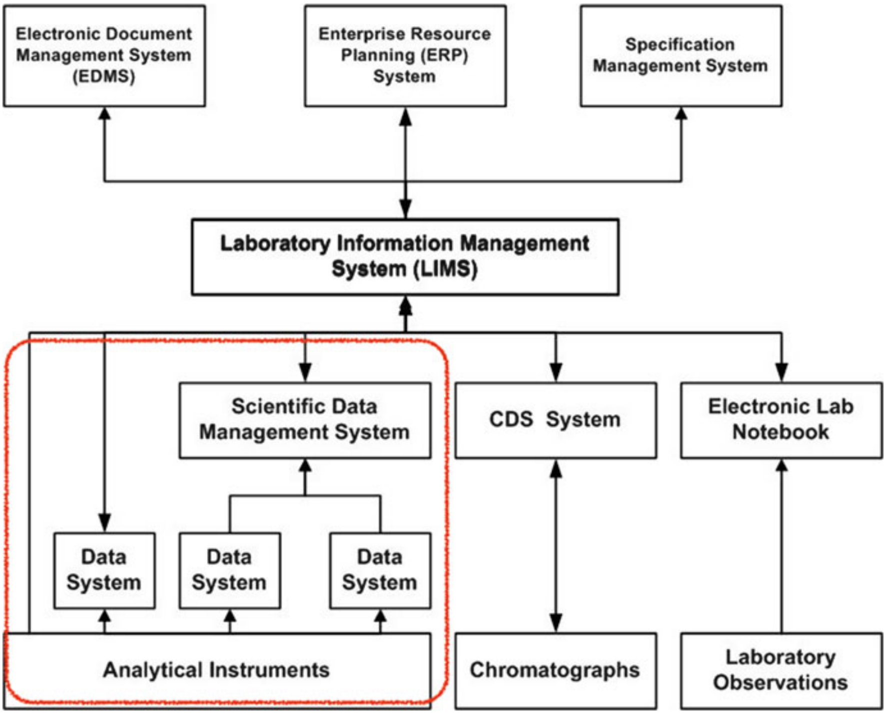


Fig. 3.1 Clinical laboratory data scheme

Once the tests to be loaded on a specific machine have been decided, a second decision problem needs to be considered, i.e. the sequencing of the tests to be conducted.

The following scheme represents a typical information based representation of a clinical laboratory. The data regarding the former problem are within the LIMS block, while the ones that characterize the latter come from the blocks highlighted by the box in the left lower side of the picture (Fig. 3.1).

During the experience described in this work, both problems were addressed, but the first one was a traditional implementation of a DSS to help manage a loading problem. Hence we focus our attention on the most peculiar and innovative part of the work: the sequencing decision problem and the attempt to embed the artificial intelligence needed to support it into existing high-throughput automated analyzers.

Generally speaking, the input to both problems is the list of the required clinical tests for each patient, along with his/her the severity index. In both cases, the general objective is to optimize the Turn-Around Time (TAT). Indeed, accuracy, precision, timeliness, and authenticity are the four pillars of efficient laboratory services. Timeliness, which is expressed as the TAT, is often used as the benchmark for laboratory performance. Fast TATs allow clinicians to obtain early diagnosis and treatment of patients and to be able to discharge patients from emergency departments sooner. Hence, faster TATs are important in general cost reduction. Delayed TATs, on the other hand, can have a bad impact on the duplication of samples sent to the laboratory, which, in its turn, leads to increased workload for the laboratory. The evaluation and improvement of turnaround times are hence important for laboratory quality management as well as to pursue patient satisfaction.

In outline, the proposed research is a show case for the use of operations research to enhance the time efficiency of high-throughput immunoassay analyzers, measured as the average time needed to obtain a test result. This goal is currently achieved by manufacturers through an intensive bio-chemical research activity mainly focused on shortening the analytical procedures. This is a *hardware-driven* approach since the deployment of faster tests usually requires the design of a new immunoassay analyzer and laboratory operating cost increase. A counterpart approach based on OR modeling is hereby presented. The proposed approach is based on a sequencing algorithm designed to optimize the average waiting time, i.e. the average time occurring between the loading on the analyzer of the test tube and the starting of the requested analytical procedure. This is a *software-driven* approach since it still shortens the average time to obtain test results but requires only that the proposed sequencing algorithm is coded and integrated within clinical Laboratory Information Systems (LIS).

The proposed optimization approach has been validated within the *Laboratorio di Analisi Cliniche Dr. P. Pignatelli* (*http://www.labpignatelli.it*), one of the most important private centers for clinical diagnosis in south Italy. All the major trademarks in the field of IVD instruments and their technology state-of-the-art are available in the *Laboratorio di Analisi Cliniche Dr. P. Pignatelli's* departments. The daily immunoassay workload of the laboratory is carried out on two Siemens Immulite®2000 analyzers, which have been considered as test case. The average waiting time saving gained by the proposed optimization method is equal to 12 min, per immunoassay, with a rise of the benefits perceived by patients equal to 8 % but without any laboratory operating costs increase. This result can be considered equivalent to the one achieved by Siemens Healthcare division with the introduction in the market of the new series of clinical analyzers Immulite®2500. This new series provides an increase in the patient benefits equal to 41.7 % but also an increase in the operating costs equal to 20 %.

The monetary benefit of the proposed approach has been also quantified in terms of the cost necessary by an alternative approach focused on obtaining the same time saving and by exploiting the clinical analyzers currently deployed. Under this scenario, an average saving of 12 min per immunoassay can be achieved only through an increase in personnel costs, amounting to 52,000 USD/year per instrument.

As discussed in the following sections, the proposed approach is tailored not only for the reference test case but also for a wide range of current high-throughput immunoassay analyzers. In this way the authors aim at providing guidelines for applying the proposed OR-based approach to other equivalent clinical analyzers.

3.2 The Technological Issues of Immunoassay Analyzers

Immunoassay analyzers are engineering systems automating an analytical process defined by a team of bio-chemists. This statement enable the possibility to describe immunoassay analyzers both from the biochemical and the engineering points of view. The biochemical point of view is related to all the analytical issues inherent in the execution of an immunoassay whereas the engineering one is related to all the issues inherent in the design of an efficient instrument to automate assays execution.

3.2.1 The Biochemical Point of View

Any immunoassay test is executed on the basis of two main sequential phases: the reaction phase and the signal measure one. During the reaction phase, a compound is obtained by adding one or more reagents to a sample of blood, urine or other bodily fluids and tissues. During the signal measure phase, the requested clinical test result is determined by measuring specific physical-chemical properties of the obtained compound. Immunoassays are well known analytical procedures which determine the presence of an analyte by specific amplification of either a chemiluminescent or a fluorescence signal. Fluorescence techniques stimulate light emission from a fluorescent label by means of a powerful excitation source such as a xenon flash or laser. Chemiluminescence-based instruments, like the reference test case Siemens Immulite®2000, stimulate light emission by adding a luminescence activator at the end of the reaction phase. Because of their overall higher sensitivity, chemiluminescence-based analyzers are more widely used than fluorescence-based instruments.

Figure 3.2 details the minimal set of sequential processing steps for a chemiluminescence immunoassay. The washing operation concludes each single incubation cycle. The overall reaction phase consists of as many incubation cycles as are the reagents to be added. The maximum number of incubation cycles in the reference case Siemens Immulite®2000 is equal to four and a single incubation cycle has a length of 30 min. Respect to the total number of assays available in the Siemens Immulite®2000 menu, 60 % of the assays lasts 60 min, 37 % lasts 30 min and only the 3 % lasts 90 or 120 min.

Once all required incubation cycles have been carried out, the signal measure starts by adding a luminescence activator (an enzymatic substrate in Siemens Immulite®2000). The luminescence activator reacts with the immunocomplex. The reaction produces light detected by a luminometer. The amount of emitted light is expressed in terms of photons per second (cps) and it is proportional to the amount of analyte in the original sample. The immunoassay result is calculated by comparing the measured cps with a cut-off value. It is worth noting that the time efficiency of an immunoassay depends on the length of the reaction phase. Indeed, the signal measure phase is based on a unique final measuring operation lasting no more than

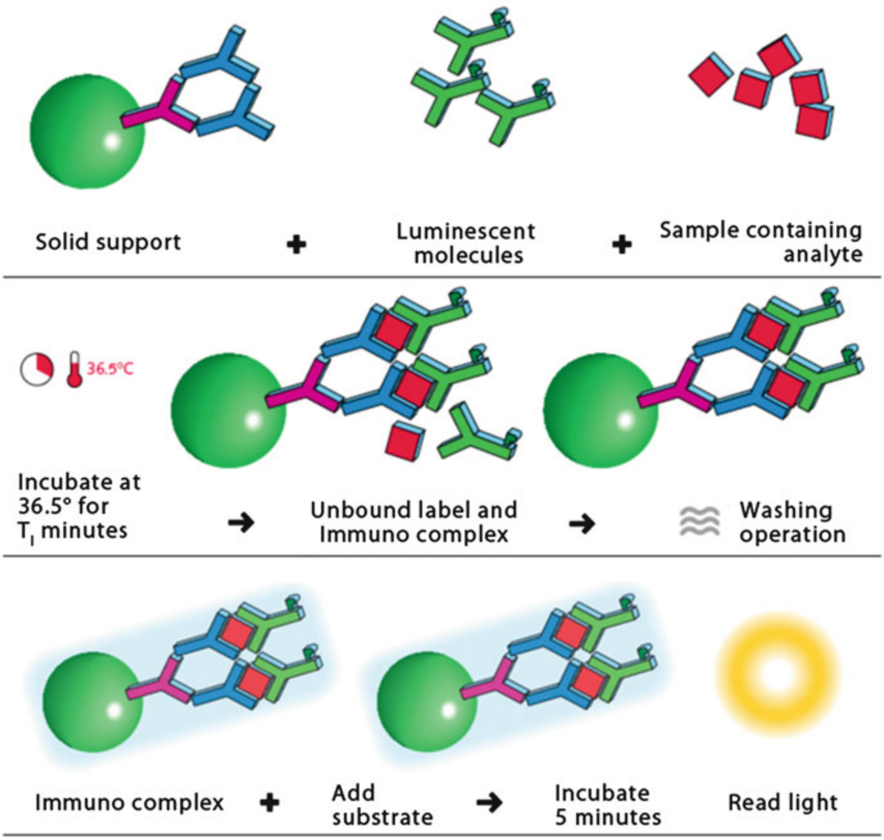


Fig. 3.2 Chemiluminescence immunoassay operations sequence

5 min, while a reaction phase might last more than 1 h. This is the issue why, in order to design faster analytical procedure, industrial research has been mainly focused on shortening the reaction phase.

3.2.2 The Engineering Point of View

In clinical laboratories, the execution of almost all IVD tests is carried out through clinical analyzers specifically designed around a set of proprietary analytical procedures. The two main elements of a clinical analyzer are the control and the analytic units. The core component of the system is the analytic unit, which in turn comprises a sampling area (① in Fig. 3.3), a reagent storage area (② in Fig. 3.3) from which the reagents are added to each sample by an automatic system (③ in Fig. 3.3), and a reaction area where the analytical procedures are carried out inside

Fig. 3.3 Immulite® 2000 XPi automated immunoassay analyzer

m reaction cuvettes (④ in Fig. 3.3). The reaction light is detected by a luminometer in the luminometric station (⑤ in Fig. 3.3).

The assays analytical workflow may differs only for the analyzer loading operation. If the sampling operation is manual, the cuvettes are grouped into a reaction plate or a disc and loaded into the reaction area. If the sampling phase is automated, the test tubes are grouped in rack and transferred to a sampling position. At the sampling position, an automatic system transfers measured volumes of samples from the test tubes into the cuvettes located in the reaction area. Each transfer is referred to as a loading operation. The automatic sampling system executes as many loading operations as are the requested immunoassays. A test tube stops at the sampling position until all the foreseen loading operations are executed.

Once a reaction cuvette has been supplied with a portion of sample, the requested analytical procedure can be carried out according to the sequence of processing steps of Fig. 3.1. The reaction phase takes place inside an incubator, which usually consists of a stepwise recirculating conveyor. Each cuvette receives the sample containing the analyte to be detected, the labeled reagent and the solid support during the first stop at the loading station. Then the first incubation cycle starts and the cuvette is moved toward the washing station through a sequence of stepwise movements of the incubator. When the cuvette reaches the washing station the current incubation cycle has been completely carried out. Once the unbounded labels have been washed out, if another incubation cycle is required the cuvette returns to the loading position (i.e. re-circulates), where a further incubation cycle can be started. For example, Siemens Immulite®2000 incubator is a winding pipeline where the cuvette circulate through a linear conveyor connecting the washing station to the loading position. If no re-circulation is required (i.e. the reaction phase is over), the luminescence activator is added and the cuvette is incubated in another conveyor connecting the washing station to the luminometer. In this case, a processing slot is freed inside the incubator and a new disposal reaction cuvette can be loaded.

In the design of clinical analyzers, system engineers hold as a key performance index the sampling throughput, defined as the number of loading operations per hour. *New and faster analytical procedures* are deployed by biochemical research teams of analyzer manufactures in order to introduce in the market *new and higher-sampling throughput analyzers*. This is the issue followed in the design of the new series Siemens Immulite®2500 that in the following will be the industrial benchmark of the proposed OR-based optimization approach.

3.3 The Operational Planning in High-Throughput Clinical Analyzers

Most of today's automated high-throughput analyzers are designed for "walk away" operations. In walk away systems, laboratory technicians only load sample tubes onto a carousel and press a start button. Consequently, at a strategic level, one of the main issue is to fit the number and the overall productivity of analytic units to the forecasted daily demand. The only constraint to satisfy is to end the test tubes loading within the manned shifts. Once instruments have been installed and integrated in the LIS, no further planning procedures are adopted. In almost all the clinical laboratory, clients are therefore served according to a *First In First Served* policy.

This approach ignores that reduction in the intra-laboratory Turn-Around-Time (TAT) in each single analyzer is achievable at an *operational* decisional. Gains may be achieved by the inclusion of optimization operations research modeling and computational techniques in the Laboratory Information System even if this opportunity is always overlooked.

Delays in intra-laboratory TAT are a major source of customer complaints and require much time and effort from laboratory staff in complaint resolution and quality service improvement. TAT is usually classified respect to four attributes: test (e.g. hemoglobin), priority (urgent or routine), population served (inpatient, outpatient, emergency department) and workflow activities (pre-analytical, analytical and post-analytical). During the last two decades, laboratory automation has appeared to be the solution for improving pre-analytical, analytical and post-analytical TAT components without increasing the number of personnel units. However, in spite of advances in clinical laboratory automation, many laboratories have had difficulty in reducing intra-laboratory TAT. A striking example is represented by point-of-care tests, i.e. the medical testing at or near the site of patient-care. Even though point-of-care tests are more costly and less precise than central laboratory services, the proliferation of point-of-care testing demonstrates that there is still a need for faster results. This probably occurs because the automation of central laboratory testing does not provide TAT improvement as a turnkey system (see [7, 14] and [5]).

In this paper we focus on analytical TAT, the most reliable and reproducible component of intra-laboratory TAT. Our reference application context is centered on routine outpatient tests, representing an important component of current disease management.

The analytical TAT component of routine outpatient tests is the time interval needed to determine the requested clinical result once the corresponding rack has been loaded onto the analyzer. The more efficient a clinical analyzer, the lower the analytical TAT interval (i.e. the lower the completion time). The analytical TAT interval is the sum of two parts. The test waiting times, during which the test tube waits to reach the sampling station and the test processing times (i.e. the time to result) during which the sampling operation and the analytical procedure are carried out. The present research aims at showing the benefits of a sequencing algorithm tailored for existing high-throughput immunoassay analyzers in order to optimize the waiting time component of analytical TAT. This approach requires no hardware change but the design of an OR-based optimization approach.

3.4 OR-Driven Solutions to Operational Planning

Prior to our OR work, the *Laboratorio di Analisi Cliniche Dr. P. Pignatelli* has already been involved as a case study in [4] focused on an OR-driven solution to minimize the analytical TAT component. In particular, the optimization approach proposed in [4] was tailored for clinical chemistry analyzers. Even if the results obtained in [4] was a remarkable saving on the average tests completion time (i.e. 25 % on average), the validation phase has been performed only via simulation. Indeed, to deploy the proposed sequencing algorithm, re-programming with a new machine cycle the system control unit would be required. Since clinical analyzer control units are not open systems, it was not possible to integrate the proposed sequencing algorithm into the planning procedures of the *Laboratorio di Analisi Cliniche Dr. P. Pignatelli*. For this reason, the optimization approach proposed in [4] should be referred to as an innovative paradigm to design new clinical chemistry analyzers rather than as a TAT optimization operating planning method. With respect to this contribution, the present research work proposes an OR-driven optimization solution which has a direct and practical impact on the daily operation of actual clinical laboratories.

3.4.1 The Proposed Optimization Algorithm: SPT^2

We start by observing that the operation planning of immunoassay analyzer may be fully represented by exploiting the taxonomy and classification schema of

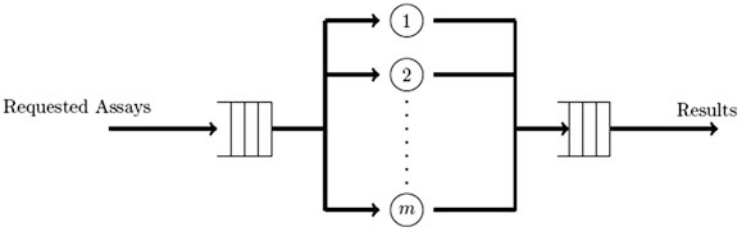


Fig. 3.4 Immunoassay clinical analyzer system layout

scheduling theory. Indeed, these instruments automatically execute a set of job types (i.e. analytes), each of them characterized by its own processing time (i.e. the time to result). In particular, all the m system processing units (i.e. incubator processing positions) are characterized by the same flow time value T_f, meant as the minimum time interval occurring between a stop at the loading station (i.e the sampling station) and a stop at the unloading station (i.e. the joint washing-luminometric stations). At each T_f, a processing unit stops at the unloading position: only if the current job has been completely processed, the processing unit is reseted (i.e. the incubator processing position is vacated and a new disposal cuvette is loaded). Since jobs (i.e the requested analytes) might last more than 1 h, system designers have to minimize the processing unit idle time, defined as the time interval occurring between the end of the job processing time and the next stop at the unloading station. For example, Siemens Immulite®2000 analyzer guarantees a null processing unit idle time interval. More generally, when the processing unit idle time is the same for all job types the loading station, the unloading station do not represent bottlenecks for the m system processing units. In this way, the layout of such automated system can be reduced as in Fig. 3.4: a bank of m identical parallel machines for which the loading station and the unloading station can be modeled as input and output unbounded-capacity buffers, respectively.

The workload of the system (i.e. the list of the requested assays) consists of n jobs partitioned into customer orders (i.e. the lists of tests requested on each test tubes). All the loading operations of each customer order must be executed consecutively. This is a batch constraint, that can be expressed as a set of precedence constraints among jobs.

The scheduling literature includes several studies on parallel machine problems with precedence constraints, as in [1–3, 6]. The most of them refer to prefixed precedences between jobs. Instead, in this case study, there is a different situation from standard precedence constraints in which one or more jobs can start if the preceding ones are concluded. The precedences required by the clinical analyzer system, not are defined in advance and they can be viewed as 'start-after-start' precedences.

More formally, we define this type of restrictions as *Generalized s-precedence constraints*. The s-precedence constraints have been introduced by Kim and Posner [9]: 'if job i s-precedence the job j, then job j cannot start processing before job i starts'. This type of constraints occur, for example, in queueing systems where earlier arriving jobs start processing first. Application can be found in many production and service industries, including health care [8]. An application case is presented in [10], where it is described an example of an automative repair shop, in which each car has several broken parts that needs to be fixed. Each mechanic is capable of fixing all broken parts, and several mechanics can work on different parts of the same car at the same time. With respect to the arrival order, cars that arrive first must start all their repair processes before cars that arrive later. The authors addressed the problem as an identical parallel machine scheduling problem of minimizing total completion time subject to s-precedence constraints. They shown that the problem is NP-hard and developed a linear programming (LP)-based list scheduling heuristic. In this paper we propose a new model in which the customer sequence (i.e. the patient arrival order) is not fixed in advance. For this reason, we believe should be introduced a new concept: the generalized s-precedence constraints. The generalized s-precedence constraints refer to all situations in which the s-precedence constraints among jobs are dependent on customer order sequence: all jobs of a customer order have to start before of all those in the next one, hence, the s-precedence constraints have to be set concurrently with the definition of the sequence order.

Minimizing the average completion time (i.e. average analytical TAT) on a high-throughput clinical analyzer can be classified by the standard scheme for scheduling problems [11] $\alpha_1|\alpha_2|\alpha_3$. The term α_1 stands for the machine structure (i.e. the analyzer layout), α_2 details the jobs characteristics or restrictive requirements (i.e. batch precedence constraints) and α_3 defines the problem objective function (i.e. the average completion time). Therefore, minimizing the average analytical TAT on a high-throughput clinical analyzer may be referred to as $P_m|g - prec|\sum C_i$, i.e the minimization of the total job completion time on parallel identical machine with generalized s-precedence constraints and job release times equal to 0. When there are no precedence constraints, it is well known in literature that the scheduling problems $P_m||\sum C_i$ is optimally solved by applying the *SPT* rule: whenever a machine is freed, the shortest job not yet processed will be started on the freed machine. However the *SPT* does not take into account any batch constraints. For this purpose, we propose to solve the considered $P_m|g - prec|\sum C_i$ problem by SPT^2, a greedy algorithm. The SPT^2 heuristic applies the *SPT* rule twice: first on customer orders and then on jobs requested on each customer order. As a result, at the loading station, the next loading operation always concerns the shortest job not yet started and requested in the customer order not yet completely *loaded* and with the shortest average processing time.

In the case of single machine, i.e. $m = 1$, the total completion time can be expressed as

$$\sum C_j = np_{(1)} + (n-1)p_{(2)} + \cdots + 2p_{(n-1)} + p_{(n)}, \tag{3.1}$$

where $p_{(j)}$ denotes the processing time of the job in the *jth* position in the sequence. Solving $1||\sum C_j$ is equivalent to assign n coefficients $n, n-1, \ldots, 1$ to n different processing times so that the sum of products is minimized. The *SPT* determines the optimal solution for the $1||\sum C_j$ by assigning the highest coefficient, n, to the smallest processing time $p_{(1)}$, the second highest coefficient, $n-1$, to the second smallest processing time $p_{(2)}$, and so on. This characterization of the optimal solution has been extended to a special case of the $P_m|g-prec|\sum C_i$

Proposition *The SPT^2 rule is optimal for the $P_m|g-prec|\sum C_i$ problem, if the following two conditions hold.*

1. *Jobs in the first m-group are processed on different machines and so on.*
2. *For each customer order, the first and the last jobs belong to the same m-group, where the first m-group ends with the m-th job in the SPT^2 solution and so on.*

Proof In a parallel machine layout, processing times can be assigned to nm coefficients: $m'n, m(n-1)'$s, ..., m ones. Solving the $P_m|g-prec|\sum C_i$ is equivalent to find a feasible assignment of a subset of these coefficients to the processing times, in such a way that the sum of products is minimized. We assume that n/m is an integer. Otherwise, we add a number of dummy jobs with zero processing times so that n/m is integer. Adding these dummy jobs does not change the SPT^2 objective function value: all these jobs are *instantaneously* processed at time zero. If the first condition holds we have that the total completion time can be expressed as:

$$\sum_{i=1}^{n} C_i = n/m(p_{(1)}+\cdots+p_{(m)})+(n/m-1)(p_{(m+1)}+\cdots+p_{(2m)})\cdots+(p_{(n-m+1)}+\cdots+p_n) \tag{3.2}$$

where $p_{(j)}$ denote the processing time of the job in the *jth* position in the sequence. Since the minimal set of the smallest coefficients are selected by the first condition, the thesis is proved if the m jobs with the largest sum of processing times have been assigned to ones, the m jobs with the second largest sum of processing times has been assigned to twos, and so on. We observe that in the $P_m|g-prec|\sum C_i$ selecting the m jobs with the largest sum of processing times is equivalent to a binary knapsack problem where the items are the customer orders, meanwhile, the weight and the worth of an item are, respectively, the number of jobs and the total processing time of a customer order.The second condition guarantees that the optimal solution to this binary knapsack problem exists and correspond to the last m jobs of the SPT^2 solution. This observations also to the second last m jobs, and so on.

In particular, we provide two sufficient conditions for detecting these special cases. The first optimality condition concerns how jobs are distributed among machines, that is if the SPT^2 solution corresponds to a schedule where the first m-group of jobs are loaded on different machines, the second m-group of jobs are loaded on different machines, and so on. The second condition checks if the SPT^2

solution corresponds to a schedule where for each customer order the first and the last jobs belong to the same m-group of jobs.

As far as all other cases where the first or the second condition do not hold we provide a lower bounding procedure to certify the quality of the SPT^2 solution.

The proposed lower bounding procedure consists in a series of splitting operations on customer orders. In particular, we supposed that each customer order o is stored as a set of jobs ordered according to the SPT rule. Each splitting operation is defined respect to a set of h breakpoint jobs $\{o[i_1], o[i_2], \dots, o[i_h]\}$, with $i_1 < i_2 < \cdots < i_h$. Each splitting operation determines a set of $h-1$ new smaller orders defined as follows: each new order includes all jobs stored in position from i_k to $i_{k+1}-1$, with $k = 1, \dots, h-1$. The first of an order is included into the set of breakpoints by default. The lower bounding procedure consists of two phases, acting to satisfy the first condition and the second condition of Proposition, respectively.

//PHASE 1

INPUT. The set O of input customer orders; the shortest job processing time D_{min}; an empty customer order set O'.

STEP 1.1. Extract a customer order o from O.

STEP 1.2. Include into the set of breakpoints of o, the job $o[i]$ such that $o[i+1] - o[i] > D_{min}$.

STEP 1.3. Split the order o according to the determined set of breakpoints.

STEP 1.4. Insert into O' the suborders obtained at **STEP 1.3**.

STEP 1.5. If O is not empty GO TO **STEP 1.1**.

//PHASE 2

STEP 2.1. Apply the SPT^2 to the set of customer orders O'.

STEP 2.2. Scan the SPT^2 solution starting from the first position. If there exists a customer order o violating the second condition of Proposition 1 GO TO **STEP 2.3**. Otherwise STOP.

STEP 2.3 Include into the set of breakpoints of o, the job placed at position i in the SPT^2 sequence such that i/m is integer.

STEP 2.4. Split the order o according to the determined set of breakpoints.

STEP 2.5. Extract the order o from the set O'. Insert into O' the suborders obtained at **STEP 2.4**.

OUTPUT. The SPT^2 solution is optimal for $P_m|g-prec|\sum C_i$ instance defined by O^{prime} and it provides a lower bound for the $P_m|g-prec|\sum C_i$ instances associated to the input order set O.

In particular the proposed lower bounding procedure relaxes a selected pool of batch constraints by splitting the corresponding costumer orders into two or more smaller orders. The goal is to obtain an instance for which the two optimality sufficient conditions hold. The splitting procedure is carried out into two steps. The first step is devoted to satisfy the first optimality condition: an order is split into a set of *suborders* where the difference between the longest job and the smallest one is not greater or equal than then the minimum job processing time of the workload. The second step concerns the second optimality condition and it requires up to n/m splitting operations. This implies that, while, in the first step, the number of splitting operations only depends on the assortment of processing times within customer orders, as far as the second step, the lower is the ratio n/m the lower the number of splitting operations. Clearly the lower are the number of splitting operations, the higher quality of the lower bound. As reported in the following section, the considered test case was almost always characterized *optimal* assortments of processing times and a low value of n/m.

3.5 Computational Results

The SPT^2 algorithm was coded and integrated into the LIS of *Laboratorio di Analisi Cliniche Dr. P. Pignatelli*. For the computational campaign, we considered all the working lists (i.e. set of customer orders) processed in the *Laboratorio di Analisi Cliniche Dr. P. Pignatelli* from November 2009 to October 2010. The total number of processed orders was on average 8400 per month, with a maximum number of requested tests per sample equal to three. The *Laboratorio di Analisi Cliniche Dr. P. Pignatelli* processes on average 400 test tubes during a height hour day with all the patients arriving between 7 am and 10 pm on weekdays. From the historical data provided, no patterns related with the observation period affect the data. The hourly trends in the number of test tubes processed and in the composition in terms of number of assays requests per sample were analyzed over all the considered weeks. No trends have been observed from the considered data and we may conclude that all the considered days are equivalent respect to the system workload. The total number of processed samples was on average 8400 per month, with a maximum number of requested tests per sample equal to three. The number of different processing times per customer orders was 1.8 on average and maximum 3.

The proposed heuristic method never exceeded 5 s of computational time and solutions were, on average, within 98 % of optimality. The quality of heuristic solutions was determined by running the lower bounding procedure. Even if special optimal cases have never occurred, about the 95 % of instances satisfied the first optimality condition. Indeed, the considered working lists were characterized by only three distinct values of processing times (30, 60 and 90 min), with the 70 % of $90'$ jobs requested as single-job orders. As far as the second optimality condition is concerned the ratio n/m never exceeded 4: the total number of jobs n was on average 400 per each considered day and maximum 410, meanwhile, the number of processing units m was 100.

The adoption of the proposed algorithm provided an average analytical TAT reduction of 12 min, by shortening the waiting time component by 12 %.

In order to evaluate which was the actual effectiveness of the two *SPT* sorting phases, we considered the two variants of SPT^2, referred to as $H1$ and $H2$. The algorithm $H1$ (i.e. $H2$) substitutes the *SPT* ordering of customer orders (i.e. intra-customer-order jobs) with the *FCFS* policy. By applying the variants $H1$ and $H2$ we obtained very different average TAT savings: few seconds with $H1$ and about

10 min with $H2$. These results clearly show that if the number of jobs within each order is very small respect to the number of machine m (in the considered test case about three jobs for each order and one hundred machines) then the optimization chances are almost determined by the customer order SPT sequencing rule. It is also interesting to quantify the industrial research effort needed to obtain an equivalent analytical TAT saving by adopting the current hardware-driven optimization approach: it is necessary to shorten by 12 min all the executable immunoassays available in the analyzers menu. In the considered case, this means that the assays lasting 30, 60 or 90 min should be reduced al least by 40, 20 and 13.3 % respectively. This type of result will be surely achieved in the future by intensifying bio-chemical research on immunoassays but at the moment it seems more as a research outlook.

In conclusion the obtained computational results show that the SPT^2 is a high-quality and efficient algorithm for optimizing the analytical TAT component at least in operating contexts similar to that one of *Laboratorio di Analisi Cliniche Dr. P. Pignatelli*. The advantages implied by the obtained analytical TAT reduction are discussed in the following section.

3.6 Quantifiable Benefits

In healthcare management any TAT reduction should be evaluated in terms of both clinical utility (i.e. benefits concerning disease outcomes) and monetary saving. These performance indicators quantify the benefits perceived by the two foremost stakeholders of a TAT reduction in a medical practice: customers (i.e. patient/physician) and diagnostic providers. By adopting the proposed OR-based approach it is possible to obtain a reduction in the assays completion time. To correctly quantify the benefits of the obtained results we first introduce, in detail, the TAT KPI and the relative metrics adopted by industrial manufacturers. Then we exploit industrial results to benchmark the proposed ones.

3.6.1 The Clinical Utility Gain

Given a work list of routine immunoassays, we define the clinical utility gain $U_{lab}(y)$ as the percentage increase of the Marginal Time Cost (*MTC*) experienced by outpatients if the average TAT is shortened by y%. Siemens Immulite®2000 and Siemens Immulite®2500 represent the input and the output, respectively, of a conventional industrial research project focused on shortening the analytical component of immunoassays TAT.

Siemens Immulite®2500 is an analyzer equivalent to and faster than the 2000 series one. Both series of analyzers are equipped to carry out an identical set of immunoassay types, among which 11 immunoassays are run faster by the 2500 series. Since Immulite®2500 has been designed to enhance surgical treatments, its main reference application context is the operating room where a work list consists of only one test. As a consequence the *MTC* of the Immulite®2500 work list can be defined as

$$MTC = \frac{(Test\ Reagent\ Cost)}{(Test\ Duration)}. \tag{3.3}$$

Since manufacturers associate a higher price to a faster reagent, the corresponding *MTC* increases as well. This means that there exists an MTC increase with respect to Immulite®2000 experienced by the in-patient when immunoassays are executed on the 2500 series. It is possible to assert that $U_{surg}(y)$, defined as

$$U_{surg}(y) = \frac{MTC \quad on \quad Immulite®2500}{MTC \quad on \quad Immulite®2000} - 1 = \frac{1.20}{1-y} - 1, \tag{3.4}$$

represents the clinical utility gain experienced by the in-patient and/or physicians if the average TAT associated to a work list of immunoassays is shortened by y% during a surgical treatment. Table 3.1 reports the utility values given to the 11 time reductions provided by Immulite®2500. The utility value gain given to Siemens Immulite®2500 in an operating room can be characterized by the value $\overline{U}_{surg}$ defined as the average value of $U_{surg}(y)$ computed over the 105 menu tests. It is worth mentioning that Siemens increases the price of all reagents of the 2500 series by 20 % respect to the 2000 ones. In particular, all the tests not included in Table 3.1 are characterized by a value of y equal to 0 and a constant utility value $U_{surg}(0)$ equal to 0.20. We interpret this result as the increased importance given to a clinical test by an inpatient under surgical treatments, who is the main end user of Immulite®2500. For these reasons, we assert that outside the operating room

Table 3.1 Time reduction in the execution of immunoassay is an important and strategic benefit for patients; for each assay we report the utility value computed on the basis of the amount of time reduction and under patient's emergency medical conditions

Medical category	Assay	Time saving (%)	Utility value
Anemia	Vitamin B12	50	1.4
	Folic acid	37.5	0.92
Cardiac markers	Creatine MB	67	2.64
	D-Dimer	67	2.64
	Myoglobin	67	2.64
	NT-proBNP	67	2.64
	C-reactive protein	67	2.64
	Troponin	67	2.64
Oncology	CA125	50	1.4
Thyroid function	PTH	67	2.64
Reproductive endocrinology	HCG	67	2.64

context, a patient gives to a y% TAT reduction a utility value $U_{lab}(y)$ equal to

$$U_{lab}(y) = \frac{1}{1-y} - 1. \tag{3.5}$$

As we did for $\overline{U}_{surg}$, the utility value gain given to the proposed optimization approach has been in outline characterized by a $\overline{U}_{lab}$ defined as the average value of $U_{lab}(y)$ computed over the actual laboratory work lists.

3.6.2 *Clinical Benefits*

The evaluation of the actual clinical benefits for a TAT saving requires the knowledge of specific clinical data: either the medical records of the patients requiring the executed assays or epidemiological data related to the diseases typically monitored through the requested test. The reference test case *Laboratorio di Analisi Cliniche Dr. P. Pignatelli* almost always does not have the use of customer medical records. Moreover, because of the wide range of diseases monitored through immunoassays, the exploitation of any epidemiological data would have been unworkable.

Nevertheless, our optimization approach is focused on shortening only the analytical component of TAT. A clinical analyzer equipped with faster reagents clearly represents an alternative optimization approach to ours.

Immunoassay analyzer Siemens Immulite®2500 is both analytical equivalent to and faster than the Immulite®2000 series. Indeed, Immulite®2500 is the result of Siemens industrial research focused on shortening the execution of a limited set of analytes (i.e. STAT-tests) already available in the Immulite®2000 assay menu. The numerical results obtained by Siemens are reported in Table 3.2.

Table 3.2 Shorten immunoassays respect to state-of-the-art of the 2000 series

Medical category	STAT-tests	STAt duration	Time saving (%)
Anemia	Vitamin B12	45′	50
	Folic acid	75′	37.5
Cardiac markers	Creatine MB	10′	67
	D-Dimer	10′	67
	Myoglobin	10′	67
	NT-proBNP	10′	67
	C-reactive protein	10′	67
	Troponin	10′	67
Oncology	CA125	30′	50
Thyroid function	PTH	10′	67
Reproductive endocrinology	HCG	30′	67

The 2500 series assay menu is composed of 105 different tests as in the 2000 series one. The Siemens Immulite®2500 analyzer is faster only on 11 assays over the complete assays menu. Since the 11 shorten STAT-tests have been designed to enhance surgical treatments, the main reference application context for Immulite®2500 is the operating room context where a working list consists of only one test. Furthermore, the reagent costs have been increased of 20 % respect to the 2000 series reagent costs. We classify the gain achieved by Siemens Immulite®2500 as hardware-driven since it is based only a reduction of the time necessary to execute an assay. On the other side, as described in the previous paragraph, by adopting the proposed OR-driven optimization approach it is possible to obtain a reduction in the assays completion time of 12 min on average with a reduction of 12 % in the average waiting time of the analytical TAT component.

In order to achieve and compare the proposed approach with the industrial benchmark, we have designed an experimental campaign based under two different scenarios cases. The first case is the one related with the daily workload of a clinical laboratory not directly linked with an hospital as in our reference case is the *Laboratorio di Analisi Cliniche Dr. P. Pignatelli*. We will refer to this case as routine out-patients case. The second case is the one related with hospitalized patients. Under this case, the immunoassays are executed under emergency conditions and this is the reason why the tests are executed in a stand-alone way (i.e. each considered working list is composed of only one sample). We will refer it as emergency in-patients case.

Under the routine out-patients case, the clinical utility gains reported in the following have been calculated taking as inputs the actual working lists (i.e. lists of sample with the relative immunoassays to be executed) processed in the *Laboratorio di Analisi Cliniche Dr. P. Pignatelli*. The average clinical utility values of both the hardware-driven and the OR-driven approaches have been computed. Table 3.3 reports the results with respect to the two different cases: emergency hospital laboratory and common clinical laboratory which usually serves ordinary patients. Case *A* refers to routine out-patient immunoassays. Case *B* refers to emergency in-patient immunoassays. Beside the clinical utility gain, Table 3.3 reports the operating cost increase. The gain obtainable by applying the proposed optimization approach to the out-patients case has been computed by taking as experimental data all the samples processed by the *Laboratorio di Analisi Cliniche Dr. P. Pignatelli's* department in the observation period. Since no trends or patterns are present in the numeric datasets, we have determined the clinical utility gain of the TAT reduction achievable on the working lists per each observation day. Otherwise, for the emergency in-patient case, we have computed the clinical utility gain under the hypothesis that each assay is processed in a stand-alone way due to the patients conditions. In this way, the clinical utility gain may be computed without the necessity to design and execute an experimental campaign but only by considering the assays available in the analyzer menu.

The proposed optimization approach (i.e. Case A—Immulite®2000), as reported in Table 3.3, provides an average clinical utility gain of 8.2 % computed on the considered working lists which is cost free from the operating point of view. This

Table 3.3 Benefits and costs of the OR approach compared with the industrial research one

Case	Analyzer series	Approach	Utility gain (%)	Cost increase (%)
A	Immulite®2000	SPT^2	8.2	0
B	Immulite®2500	Shorter assays	41.7	20
A	Immulite®2500	Shorter assays	2.1	20
B	Immulite®2000	SPT^2	0	0

result may be considered equivalent to the current industrial research (i.e. Case B—Immulite®2500) characterized by an average clinical utility gain of 41.7 % and an operating cost increase of 20 %.

The row Case A—Immulite®2500 reports the results obtainable by executing the considered work lists onto the Immulite®2500. In this case, a clinical utility gain of 2.1 % reflects that *Laboratorio di Analisi Cliniche Dr. P. Pignatelli's* departments deal with out-patient routine immunoassays among which only 15 % belong to the set of shortened tests of Table 3.2.

Finally, the row Case B—Immulite®2000 points out that the proposed method does not provide any TAT saving if applied to emergency in-patient work lists.

In conclusion, the innovation provided by Siemens with the introduction in the market of the new series Immulite®2500 is directly driven for the operating room context. In this application context the clinical utility gain given to a TAT saving is clearly different from that one given by out-patients even if the number of tests executed per day under emergency conditions is significantly lower respect to the routine out-patients case. The hardware innovation proposed by Siemems gives low advantages if applied to clinical laboratory with an increase in the daily operating cost. Otherwise, the proposed OR-approach that may be directly integrated in any Laboratory Information System outperforms the Siemens result in term of clinical utility gain perceived by out-patients without any increase in laboratory operation costs.

3.6.3 Monetary Benefits

The possibility of improving intra-laboratory TAT without any operating cost increase is a relevant result. In a time of global economic instability and continual pressure to reduce healthcare costs for most clinical laboratories the only way to follow current healthcare trends is to do more with the same or even fewer resources.

Table 3.3 shows that by leveraging on existing laboratory automation, TAT can be optimized by an operations research-based approach (i.e. Case A—Immulite®2000). A reduction of intra-laboratory TAT by 12 min is not only a cost-free result for the diagnostic provider. Indeed, alternative approaches not involving laboratory automation can be accomplished only by increasing personnel, the other main laboratory resource. In the *Laboratorio di Analisi Cliniche Dr. P. Pignatelli*, the lowest-automated but still highest manned TAT activity is ordering: patients are queued for registration and sampling. In order to obtain an equivalent saving *Laboratorio di Analisi Cliniche Dr. P. Pignatelli* may directly acts only on the ordering phase by increasing the personnel units. The corresponding personnel cost increase would be 52,000 USD/year, which has been considered as the monetary benefit provided by the proposed optimization approach.

3.7 System Design and Development

3.7.1 Class Diagram

A static view of the building blocks of the DSS is depicted in the class diagram in Fig. 3.5. The diagram includes four classes: Batch, Job, Machine and Scheduler, which are illustrative of the main entities involved in a classical scheduling problem. As stated in Sect. 3.4.1, the operation planning of a clinical analyzer can be modeled according to the taxonomy and the classification schema of scheduling theory.

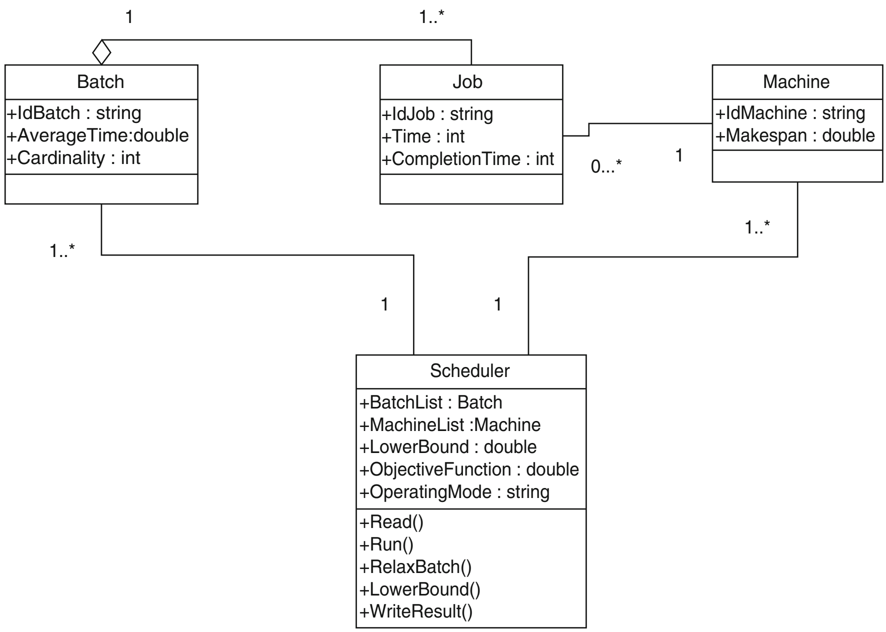


Fig. 3.5 Class diagram

3.7.1.1 Batch

The Batch class is associated to the concept of customer order (patient test tube) that requires a subset of jobs (analytes). The batch is identified by the attribute IdBatch and it is characterized by the number (Cardinality) and by the average time (AverageTime) of jobs included in it.

3.7.1.2 Job

The Job class denotes a single task, i.e., a requested assay. Each job having a given processing time (Time), is identified by the attribute IdJob. The CompletionTime attribute refers to the analytical TAT, i.e. the time when the test analytical procedure ends. Batch and Job classes are interrelated to each other through the aggregation relation. It means that the Batch object is made up of one or more Job objects, according to the multiplicities indicated on the corresponding link.

3.7.1.3 Machine

The Machine class represents the resource type of the system that processes the jobs. In this case, as described in previous section, the Machine class models the incubator processing position that it is identified by the attribute IdMachine. After the sequencing order has been defined, each machine presents an own makespan value (Makespan) i.e., the total time spent by the machine in performing its daily workload. The association between Job and Machine class states that one Job can be assigned to one and only one machine. On the other hand, zero or more jobs can be sequenced on each machine, that can handle one job at a time.

3.7.1.4 Scheduler

The Scheduler class is the entity that defines the overall schedule of workload. In the context of Clinical Analyzer, it deals with the operation planning in term of sequencing of the required analytes at the loading station. The scheduler is characterized by the list of jobs involved (BatchList) and by the available machine set (MachineList). It get the input data by means the reading procedure, Read. Furthermore, the method Run determines the job schedule and hence it set the objective function value (ObjectiveFunction), by computing the job total completion time. There are different operating modes that are defined by the attribute OperatingMode that indicates the heuristic to be executed by the Run procedure, SPT^2, $H1$, or $H2$. In order to evaluate the solution quality, it is possible to consider the relaxed problem. For this scope, a relaxation of batch composition is required and it is performed by the RelaxBatch procedure; hence, the lower bound value (LowerBound) is obtained by applying the corresponding LowerBound method. Finally, the results are saved on text file

(WriteResult method). The class Scheduler is logically associated with Batch and Machine objects by two suitable connections. These links indicate that the solver expects at least one batch and one machine to identify a feasible schedule.

3.7.2 Activity Diagram

The DSS was been integrated within the LIS. The activity diagram reported in Fig. 3.6 displays the sequence of activities and identifies the actors responsible for each action.The activities start with the selection of workload (the patient list) to plane by the health decision maker. Next, the LIS sets parameters of the Dss application: paths of input files and the operating mode. The action flow passes to DSS actor, by the Launch Dss action. In order to get the input data to process, Launch Load action is used by the Dss to read and load data into the Dss software structures. At this stage, DSS launches the scheduler (Launch Schedule), according to the operating mode selected by the LIS actor, and once the job sequence is defined, the DSS creates the file of output results (Generate Output File). Once the solution is available, the operation flow, forks into two parallel branches. One branch is related to the Clinical Analyzer Operator actor: the Clinical analyzer interface shows a set of indicators which allows the operator to analyze (Evaluate KPI) some performance indicators and decide whether there are any critical issues to report (Alert).

From LIS point of view (the other branch), the decision maker, evaluates the solution and he can decide to:

- accept the schedule of assays provided by DSS and send the work sequence to the clinical analyzer (Plan Operations)
- Do not accept the Dss solution: in this case there are two additional possibilities:
  - The decision maker processes and formulates an own sequence and then send it (Plans Operations) to the clinical analyzer;
  - A new run of Dss can be started by considering new processing settings.

3.7.3 User Interface

The DSS is intended to be embedded inside existing high-throughput clinical analyzers, which come with preinstalled UIs like the one reported in Fig. 3.7.

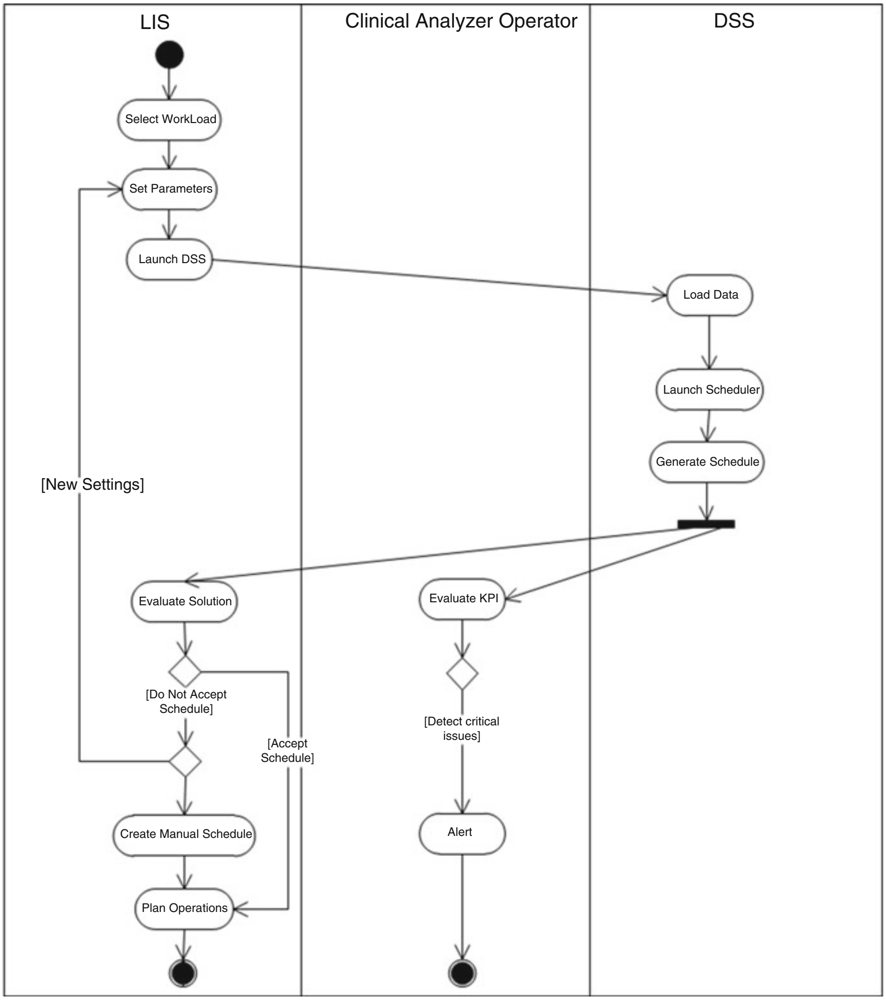


Fig. 3.6 Activity diagram

3.7.4 Classification

According to Power's [12] and [13] expanded DSS framework, the DSS case study we present is an example of the model-driven DSS type.

- the *dominant DSS component* is an optimization heuristic approach based on the well known SPT rule, which acts as the *enabling technology*;
- the *target users* are the clinical analyzers supervisors;
- the *purpose* of the DSS is to allow both an automation and an optimization of a relevant part of the planning and scheduling process;

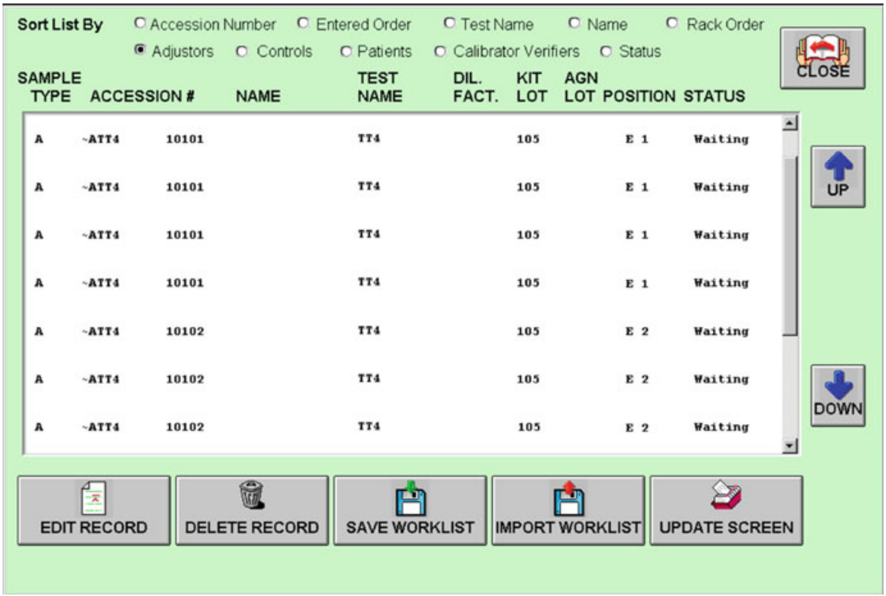


Fig. 3.7 A clinical analyzer UI

- the targeted deployment technology was to develop a decision support system embedded into existing automated clinical analyzers, but the attempt was not successful, so the actual implementation remained at a prototypal stage as a C# implementation of the proposed algorithm.

3.8 Lessons Learned

Despite the proposed approach proved to be extremely effective, our attempt to propose it to the producers of automated clinical analyzers failed. We tried to propose our results to determine guidelines for enhancing a wide range of current high-throughput clinical analyzers, but we hit a wall trying to convince technology providers to invest in embedding such new models in their hardware. In this section we summarize, in a question/answer form, the most important lessons we learned from this experience.

Is it worth trying to spend research time to develop algorithms for automated systems already on the market?

The experience reported with this case study shows how it is often possible to improve the performances of complex automated systems, in this case applied to clinical analyzers but using techniques that can be easily adapted to other industrial sectors. Automated systems are getting more and more integrated, also dealing with efficient energy usage aspects, as well as system's size and usability. The complexity of architectures and components necessarily leads to such complex aspects in the operations management, that only innovative approaches in terms of models and algorithms can be able to address.

Internet of Things (IoT) is one of the main trends in technology related research. The recent evolution in the technologies that can take advantage of the IoT will lead to a great need for models and algorithms that allow the optimization of decisions taken as a result of this never seen before availability of information in a structured and digital manner. New research streams in this sector can be easily forecasted in the next years.

The case study we presented took its start from more traditional connected sensors within an already existing automated system, but we can expect that the data flow that feeds the method we proposed will easily be provided by the IoT in the next future. Investing research time in this field, hence, proves to be a good investment as a research inspiration in the next years.

How should a University protect the research results to facilitate the transfer to automated systems producers? Are international patents the right tools?

Universities will have to intensify the development of innovative methods on one hand, while, on the other hand, companies will need to coordinate the release of open protocols and architectures to allow the effective usage of the larger and larger amounts of data that production systems will provide.

The success of such technologies that will manage a large amount of automation will shift the focus from abstract decision methods to concrete and complete hardware/software solutions. In this scenario, international patents that cover the whole solutions could be a prolific territory for conjunct patents between Universities and companies.

Can the local/national boundaries of a University be a limitation to the diffusion of its research results when this is applied to commercial automated systems?

The proposed work was presented by the Italian University of Salento to Roche Italy. The attempt to spread the results to a wider, at least European, audience through this channel failed.

According to our experience, being part of a relatively small research institute in Italy was a strong limitation to the diffusion of our research results.

Was the presentation modality the cause of the failure in the attempt to transfer results to technology producers? Why? How can this be avoided?

Trying to present the research results through a direct channel was not effective in our case. The scientific publication and conference channel, as well, led to no progress in the attempt to let the success story inspire further usage of the achieved results.

Find the right diffusion channel and possibly presenting the results under different points of view constitutes one of the main objectives of our future work on this subject.

Will the Industry 4.0 model diffusion help the transfer of academic research to technology producers?

The diffusion of IoT and Internet Everywhere technologies will provide a great push to the development of automatic methods to perform optimal decisions when large amounts of data can be used to determine it. In this sense, the proposed study can be seen as a precursor work in this area.

3.9 Conclusions

Today, IVDs account for less than 3 % of healthcare costs. Yet, it provides nearly 80 % of the information used for therapy selection. The decline in skilled resources has driven the transformation toward full laboratory automation. While there is an attraction to new technologies, the time-to-result reduction challenge is tackled by the IVD industry with a hardware-driven approach. In this paper, we have shown how elements of operations research can be exploited to give a *soft* answer to the TAT reduction challenge.

The proposed approach has been validated on the Siemens Immulite®2000 high-throughput clinical analyzer. The analyzer has been modeled as a parallel machine system and a sequencing algorithm has been designed to minimize the test tube average completion time. This algorithm has been coded and integrated into the Laboratory Information System of the *Laboratorio di Analisi Cliniche Dr. P. Pignatelli*, where its evaluation has demonstrated the success of a *soft* operations research-based approach to reduce time-to-results in clinical practice.

Unfortunately, trying to extend this success story to the producers of clinical analyzers did not turn into a success due to several issues that we discussed in the chapter.

Acknowledgement This work was supported by Progetto PON/MIUR Vis4Factory: Sistemi Informativi Visuali per i processi di fabbrica nel settore dei trasporti—Codice Progetto: PON02 00634 3551288.

References

1. Aho, I., Mäkinen, E.: On a parallel machine scheduling problem with precedence constraints. J. Sched. **9**(5), 493–495 (2006)
2. Chekuri, C., Motwani, R., Natarajan, B., Stein, C.: Approximation techniques for average completion time scheduling. SIAM J. Comput. **31**(1), 146–166 (2001)
3. Gacias, B., Artigues, C., Lopez, P.: Parallel machine scheduling with precedence constraints and setup times. Comput. Oper. Res. **37**(12), 2141–2151 (2010)
4. Grieco, A., Guerriero, E., Musmanno, R., Tolio, T., Anglani, A.: Scheduling in dial-indexed production lines. Int. J. Prod. Res. **41**(14), 3139–3158 (2003)

5. Hilborne, L.H., Oye, R.K., McArdle, J.E., Repinski, J.A., Rodgerson, D.O.: Evaluation of stat and routine turnaround times as a component of laboratory quality. Am. J. Clin. Pathol. **91**(3), 331–335 (1989)
6. Hoitomt, D.J., Luh, P.B., Max, E., Pattipati, K.R.: Scheduling jobs with simple precedence constraints on parallel machines. IEEE Control. Syst. Mag. **10**(2), 34–40 (1990)
7. Howanitz, P.J., Cembrowski, G.S., Steindel, S.J., Long, T.A.: Physician goals and laboratory test turnaround times. Arch. Pathol. Lab. Med. **117**, 22–28 (1993)
8. Kim, E.-S.: Scheduling of uniform parallel machines with s-precendece constraints. Math. Comput. Model. **54**, 576–583 (2011)
9. Kim, E.-S., Posner, M.C.: Parallel machine scheduling with s-precedence constraints. IIE Trans. **42**, 535–537 (2010)
10. Kim, E.-S., Sung, C.S., Lee, I.S.: Scheduling of parallel machines to minimize the total completion time subject to s-precedence constraints. Comput. Oper. Res. **36**, 698–710 (2009)
11. Pinedo, M.L.: Scheduling: Theory, Algorithms, and Systems. Springer Science & Business Media, Berlin (2012)
12. Power, D.J.: Supporting decision-makers: an expanded framework. In: Proceedings of Informing Science and IT Education , 1 June 2001, pp. 1901–1915 (2001)
13. Power, D.J.: Decision Support Systems: Concepts and Resources for Managers. Quorum Books, Westport (2002)
14. Steindel, S.J., Howanitz, P.J.: Physician satisfaction and emergency department laboratory test turnaround time: observations based on college of American pathologists q-probes studies. Arch. Pathol. Lab. Med. **125**(7), 863–871 (2001)

Chapter 4
Decision Support Systems for Energy Production Optimization and Network Design in District Heating Applications

Andrea Bettinelli, Angelo Gordini, Alessandra Laghi, Tiziano Parriani, Matteo Pozzi, and Daniele Vigo

Abstract This chapter focuses on the application of Operations Research approaches to energy distribution and production. As to the energy distribution, we consider the strategic problem of optimally expanding a district heating network to maximize the net present value. As to the energy production, the problem we consider is at a tactical level and concerns the definition of the best unit commitment plan to maximize the profit of the plant. The chapter presents two decision support systems designed to address the above mentioned problems and discusses their use in real-world applications.

4.1 Introduction

Energy production from fossil energy sources, which has been the boost for the industrial development of the second half of the Nineteenth Century, has developed a dangerous dependence on oil producing Countries that the European Union is trying to reduce. The European strategy counts on energy production from renewable energy sources leading to a partial (sometimes total) self-sufficiency, at the same time lowering greenhouse gas emissions. The future development of energy production and distribution relies on the careful exploitation of different, renewable energy sources, each one with its own costs and profits, sometimes dependent on weather and/or daylight hours (e.g., thermal solar). Looking at

A. Bettinelli • A. Gordini • T. Parriani • M. Pozzi
OPTIT Srl, Viale Amendola 56/D, 40026 Imola (BO), Italy
e-mail: andrea.bettinelli@optit.net; angelo.gordini@optit.net; tiziano.parriani@optit.net; matteo.pozzi@optit.net

A. Laghi (✉) • D. Vigo
University of Bologna - Department of Electrical, Electronic and Information Engineering "G. Marconi", viale Risorgimento, 2, 40136 Bologna, Italy
e-mail: alessandra.laghi2@unibo.it; daniele.vigo@unibo.it

© Springer International Publishing Switzerland 2016
J. Papathanasiou et al. (eds.), *Real-World Decision Support Systems*, Integrated Series in Information Systems 37,
DOI 10.1007/978-3-319-43916-7_4

the Italian arena, the annual publication of AIRU (the Italian Association of Energy Utilities) shows that, at the end of 2014, the most common type of energy production plants connected to district heating networks (boilers excluded) was Combined Heat and Power (hereafter CHP) fueled by carbon sources along with electric power plants. The latter, even if principally aimed at electricity production, become thermal production plants when the heat produced is recovered to heat a district heating network. The number of plants producing heat and electricity from waste incineration is growing, as well as the number of those using biofuels. Other energy sources are geothermal, heat pumps and heat waste recovery from industrial processes. In Italy there is only one solar thermal plant connected to a district heating network, opened in May 2015 near Varese (northern Italy).

From a management standpoint, energy production and distribution are both challenging business issues for (multi)utility companies, and have different perspectives:

- distribution is a strategic business issue, related to the design of the district heating network, that can be either the extension of an existing network or the design of a completely new network from scratch; network-related decisions require huge investments, due to the cost of materials and civil works for the realization of the network, thus the board of management aims at maximizing the Net Present Value (NPV) of the investment.
- Production is a tactical issue, because the management cycle has a yearly horizon and is defined by the budget and (at least one) revised budget; in this business case, the aim is the maximization of EBITDA (Earnings Before Interest, Taxes, Depreciation, and Amortization). In both such business issues, traditional (manual) management becomes weaker and weaker as the complexity of the plant/network grows. Hence, Decision Support Systems (DSS) are necessary to support decision makers in finding the optimal solution.

Given these introductory considerations, this paper will present a suite, developed by Optit Srl in collaboration with the University of Bologna, of two DSSs for tackling both network design problems (OptiTLR) and energy-production management problems (OptiEPM). The two applications are described in the two following sections: Sect. 4.2 describes the DSS for network design problems, and Sect. 4.3 is focused on energy production management. Both sections are identically structured: after a presentation of the problem and an overview of the related literature, system requirement analysis and system design are described. Then follows the discussion of system development and user interface design, the description of the optimization module and of the system user experience. Conclusions are drawn in Sect. 4.4 for both DSSs.

4.2 Business Issue n. 1: District Heating Network Design

The first business issue, focused on the district heating network itself, is about strategic planning. The main questions to be answered are the following:

- commercial development: given a district heating grid and its current users, who are the best new users to encourage to join in order to maximize the NPV of the system, respecting its thermo-hydraulic constraints? The same issue can be considered where there is a saturated grid: if there are users that do not completely exploit the contractual power, would it be advantageous to change their contracts and make part of their power available for new users, without expanding the grid?
- strategic development: given two or more grid expansion configurations (e.g., new backbones), and a set of prospects, which subset maximizes the NPV?

In this business issue, commercial drivers must find a compromise with technical constraints (i.e., with the thermo-hydraulic consistency of the network).

At the moment, manual management relies on thermo-hydraulic simulation tools that allow network managers to find feasible solutions. This approach has two strong limitations. First of all, simulation tools do not perform optimization, so network managers can simulate different configurations of the network but the comparison has to be made out of the system. Secondly, the commercial point of view is absent in these simulation tools, so the evaluation of the best commercial alternative among the physically feasible ones has to be made using fact sheets.

We are not aware of any other commercial solution addressing optimal design of networks; simulation tools are definitely available but, as previously mentioned, the optimization perspective is absent. Examples of simulation tools are InfoWorks ICM by HR Wallingford[1] and Marte AQVA by DEK S.r.l.[2]

Referring to the classification proposed in the introductory chapter, OptiTLR is a Model Driven DSS, since the dominant component is a quantitative model; targeted users are network managers and their staff. The purpose of OptiTLR is to create optimized possible expansions of the network; the enabling technology is an open source Geographical Information System (GIS), whose OptiTLR is a plug-in.

4.2.1 Literature

Research on representation and simulation in details of the behavior of the thermo-hydraulic network through sets of non-linear equations can be found in literature, for example [2] and [18]. Solving systems of non-linear equations is difficult and

[1] http://www.hrwallingford.it/.

[2] http://www.dek.it/.

computationally expensive. For this reason, aggregation techniques of the network elements are often used to model large district heating networks, at the expense of some accuracy [10, 11, 13, 14, 23, 24].

In [1], an integer programming model is proposed for the optimal selection of the type of heat exchangers to be installed at the users' premises in order to optimize the return temperature at the plant. The authors achieve good system efficiency at a reasonable cost.

Finally, Bordin et al. [3] develop a mathematical model to support district heating system planning by identifying the most advantageous subset of new users that should be connected to an existing network.

4.2.2 System Requirements Analysis and System Design

The use of OptiTLR requires the configuration both of the existing network, if any, and of the prospects. The tool can be also used for defining a completely new network starting from scratch. Furthermore, it requires the definition of a few optimization parameters. As for the network configuration, the following data is needed:

- existing network:
  - cartographic data showing backbones, branches, nodes, roads, buildings, barriers;
  - plants of the network: location and configuration;
  - customers: location, contractual power, heat usage, tariffs.
- Potential expansions:
  - prospects: location, expected required power;
  - backbones: layout;
  - connections: automatic definition modifiable manually by the user;
- Costs and revenues:
  - Implementation costs
  - tariffs
  - acquisition curve
  - amortization period

As to the optimization parameters, three categories of parameters have to be set:

- technical, such as pressure drop, concurrent factor;
- economical, such as maximum number of new customers, maximum investment;
- contractual, such as tariffs of prospects.

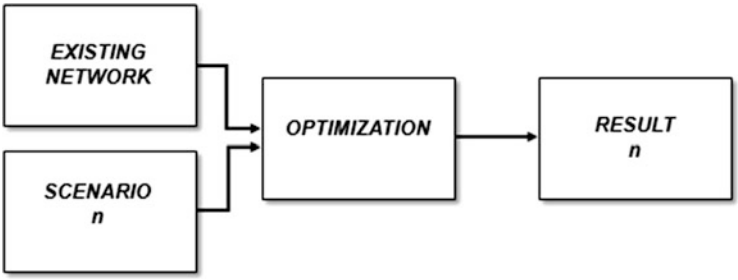


Fig. 4.1 Diagram of OptiTLR rationale

Figure 4.1 shows the rationale of the tool. The user can generate as many alternative configurations as needed by changing input parameters. The generation of results is rather quick (about 20 configurations in 4 h, allowing the user to perform an accurate what-if analysis); the same activity would take much longer using traditional tools!

According to our computational experience, solving the model directly (monolithic) is possible only for small real-world instances: for networks with about 400–500 nodes, it requires a typically 500 s. For large instances the monolithic approach does not even find a feasible solution within the time limit of 10 min. Our 3-phases method instead is able to produce high quality solutions for all the instances we typically consider. Moreover, the small instance is solved to optimality in shorter computing time (i.e. about 200 s) with respect to the monolithic approach.

4.2.3 System Development and User Interface Design

OptiTLR is a desktop application, whereas the optimization module and the solver are on a server managed by Optit; therefore, there are no IT integration requirements.

To access the application, technical and commercial profiles can be set up. Figure 4.2 shows an example of user interface: OptiTLR is a plug in of the open source GIS OpenJump, an open source GIS based on Java that, according to our experience, proved considerably user-friendly and efficient in the treatment of large amounts of vector data. Therefore the overall look-and-feel resembles GIS tools, with a big window showing maps, a left-hand column for managing the tree of layers and a menu at the top with functional buttons. All the functionalities of a GIS tool are therefore available, especially the manual drawing feature.

For commercial users, a detailed report is generated, showing the best prospects to be connected (and the corresponding contractual power), customers to whom a new contract could be proposed (if any) and detailed costs and revenues.

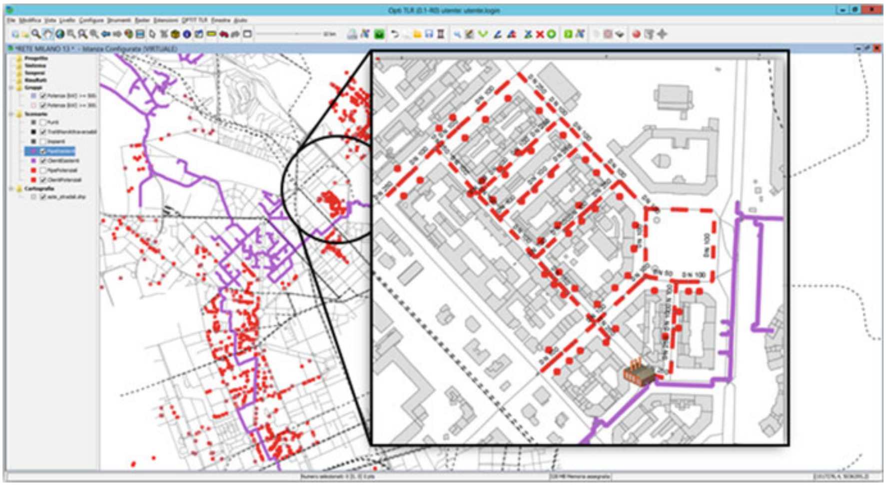


Fig. 4.2 Example of OptiTLR user interface

4.2.4 *Optimization Module*

The optimization module is hosted on a server by Optit, as mentioned above.

We model the problem as a Mixed Integer Linear Program (MILP). It is an extension of the model proposed in [3].

The problem is to define the extension plan for a district heating network that maximizes the NPV at a given time horizon. It is therefore necessary to decide: (1) the set of potential new customers that should be reached, (2) which new links should be installed, and (3) their diameter. Both an economic model and a thermo-hydraulic model have to be considered. The economic model takes into account:

- production cost and selling revenues,
- cost of network link activation, that depends on the diameter of the selected pipes,
- cost of customer connections,
- amortization,
- taxes,
- budget constraints.

Moreover, while the investment on the backbone link is done in the first year, new customers are not connected immediately, but following an estimated acquisition curve (e.g., 25 % the first year, 15 % the second year,. . .). Hence, the values of the corresponding costs and revenues have to be actualized accordingly. The thermo-hydraulic model must ensure the proper operation of the extended network. The following constraints are imposed:

- flow conservation at the nodes of the network,
- minimum and maximum pressures at the nodes,

- plants operation limit: maximum pressure on the feed line, minimum pressure on the return line, minimum and maximum flow rate,
- maximum pressure drop along the links,
- maximum water speed and pressure drop per meter.

District heating networks can be quite large (hundreds of existing and potential users, thousands of links) making it difficult to solve the problem directly with the full MILP. For this reason, we start with a restricted problem, where some of the decision variables have been fixed, and solve it to optimality. Then, some variables are freed and the process is iterated until either the time limit is reached or the unrestricted problem has been solved (i.e., we have reached a global optimal solution).

4.2.5 System User Experience

OptiTLR has been successfully used by two of the largest Italian multi-utility companies. As an example of successful application, we can refer to the following: recently, in a large Italian town, a transport pipeline has been built, interconnecting two previously independent networks, which allows significant improvements to heat dispatching policies, but also paves the way for great opportunities in the implementation of a larger expansion of the network in a densely populated urban area. The resulting system has a huge heat demand potential: the area across the new line accounts by itself for more than 100 MW in less than 1 km width. Meanwhile there are limiting conditions on the production side and for transport capacity of the system. For that reason, it was necessary to analyze and compare a lot of small areas to decide on the DH development plan. Usually, on the basis of the potential customers, technicians have to define the new pipes and calculate costs, by using the hydraulic analysis tool if necessary. Then the financial department prepares the discounted cash flow, with the aim to maximize the investment rate of return. To reach this goal, they try different network configurations (e.g. by adding or cutting branches), so the process may require some time. OptiTLR has enabled significant improvements in the capability to generate, analyze and design cost-effective development scenarios. Namely, the main key success factors of this application are:

- **quick generation of results:** users are allowed to generate and compare as many different network configurations as they want in a relatively short period of time (about five new configurations per hour). This extremely simplifies what-if analysis activities and the decision process itself.
- **Interconnection between two different perspectives:** commercial staff and technicians find in the same tool a support for their respective activities, each one getting results compliant with the other's goals/constraints.
- **High number of decision variables** within the model, allowing decision makers to perform many kinds of what-if analyses.

4.3 Business Issue n. 2: Energy Production Management

This business issue is about a plant manager who has to serve one or more demands (heat demand mainly, but also cool demand and/or electricity demand if the plant is connected to a building or to an industrial facility) and these demands can be satisfied by producing energy using different machines, depending on the plant configuration. So the goal of a plant manager is to satisfy the customers' demands by choosing the best energy production mix to maximize the EBITDA. Consequently, the decision-making process must take into account the following factors:

- costs, profits and fiscal advantages (if any) of each energy source;
- technical constraints of the plant itself and of the machines;
- regulatory constraints;
- ordinary and extraordinary maintenance requirements.

Also, if the electricity produced by the plant is sold to the National Electricity Network, there is an additional variable (the selling price) and an additional constraint (the amount of electricity committed to the market the day before for the following day).

Manual management relies on several fact sheets to manage the typical business processes:

- **budgeting**, to define the overall energy production for the year to come;
- **revised budgeting**, to modify the budget objectives on the basis of the results of (at least) the first semester;
- **weekly operations**, most of the times based on the experience of the plant manager (in respect of the previous years) and defined on a monthly basis.

As such, decision making for energy production is a complex job, and this complexity increases for bigger plants and if more production alternatives are offered.

Similar DSSs of this kind already exist and can be classified into two categories:

1. academic demonstrators: tools developed by universities or research centers that address the optimization of energy production; usually these tools model the problem very well but lack "industrial quality", that means that the user interface is poor and does not address all the business needs of a plant manager;
2. industrial DSSs for energy production optimization: this is the category our DSS belongs to; the added value of OptiEPM compared to its competitors is the careful modeling of Country-specific regulatory requirements such as, in the Italian case, different fiscal incentives on gas depending on the final use, or the use of the "virtual machine" defined in Italian legislation if the yearly mean performance of the CHP engine is less than 75 %. Examples of DSSs for energy

production optimization on the market are energyPRO by EMD International,[3] UNICORN by N-SIDE,[4] ResOpt by KISTERS.[5]

Referring to the classification proposed in the introductory chapter, OptiEPM is a Model-Driven DSS, since the dominant component is a quantitative model; targeted users are plant managers and their staff, each one having their own system profiles to perform only allowed actions. The purpose of OptiEPM is to determine the optimized production plan (i.e., machines scheduling), and the enabling technology is MILP combined with a relational database for the input variables.

4.3.1 Literature

The OptiEPM underlying MILP is a particular case of the well known Unit Commitment (UC) problem (see, e.g. [17, 22]), in which the goal is to determine how to dispatch the committed units in order to meet the demands and other constraints cost-efficiently. UC problems, even when CHP is not considered, are complex in practice. The solution methods used to solve UC problems span from Lagrangian relaxation, as proposed in [4, 12], to genetic [8] or tabu search [15] heuristics. Attention has been put as well on obtaining efficient MILP formulations (see e.g. [6, 16]).

Interdependencies between power and heat productions make realistics CH and P power units even more difficult to be optimized [20]. In some cases general purpose MILP techniques are applied, such as the Branch and Bound (see [7]). Resolution methods may otherwise rely on time-based decomposition, as in [9], dynamic programming [19] or again Lagrangian relaxation [21].

4.3.2 System Requirements Analysis and System Design

To use OptiEPM, it is necessary to set both the plant configuration and the input data needed to forecast the energy demands.

As for the plant configuration, the following requirements have to be gathered:

- **final use of the energy produced by the plant**, to distinguish whether the plant is connected to a district heating network or to a single user (i.e., energy service contract).

[3] http://www.emd.dk/.

[4] http://energy.n-side.com/.

[5] http://www.kisters.net/.

- **Technical characteristics of the machines**, to set the type of machines, their status (active/inactive), production performance and operating levels (minimum, maximum and steps).
- **Economic variables, to set fuel taxes**, white/grey/green certificates price and cap (if any), other incentives, maintenance costs, energy selling price, buying price of fuel and electricity.
- **Regulatory constraints**, to set any yearly constraints on the ratio between heat and electricity production or other constraints.

As for the input data necessary to feed the forecasting model for the energy demands, two flows of data are required:

- **weather data** (temperature forecast for one/two days later and observed temperature for the day before);
- **production data**, that means the historical series of energy production.

OptiEPM system design is represented in Fig. 4.3: there is a forecasting module, to predict the demands of the energy related to the specific plant (i.e., heat, cool and electricity); this module needs as input the historical series of daily temperatures and the historical series of production for each type of energy. The forecasted demands serve as input for the optimization module, along with the plant configuration, selling prices for energy vectors and buying prices for gas and electricity. Given these inputs, the optimization tactical planning module defines the yearly budget and estimates the optimal allocation of yearly requirements.

Then the operations module runs to optimize the scheduling of machines, along with the weather forecast (usually 4 or 5 days ahead). Every month (or quarterly, depending on reporting frequency) a set of key performance indicators (KPI) can be

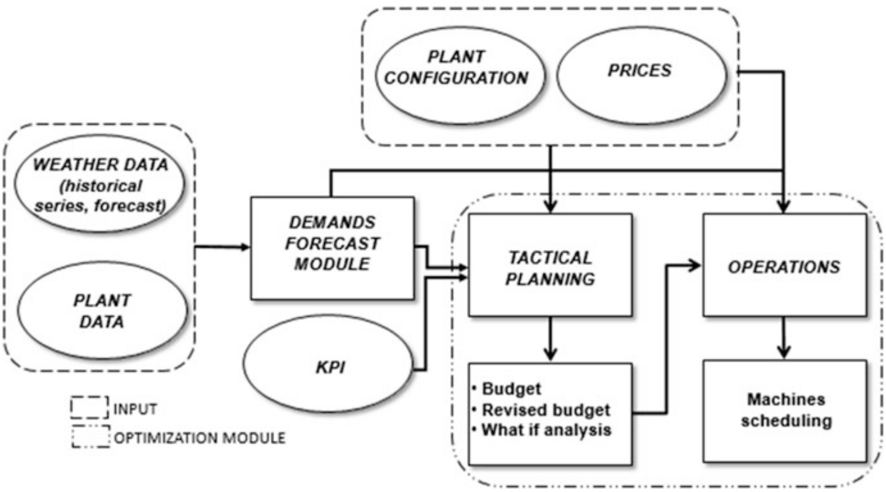


Fig. 4.3 OptiEPM system design

uploaded into the DSS to revise the budget and to have a more accurate estimate of the remaining share of yearly requirements.

4.3.3 System Development and User Interface Design

OptiEPM solution has been developed using a Google Web Toolkit and Enterprise Java Beans 3 framework and is available as software-as-a-service, accessible by a web-browser. As such, users can access the application by using their browser and no other IT requirements are needed. System integration, necessary for input data, is achieved through an FTP folder where every day comma separated value files are uploaded by the utility itself by means of an automatic procedure.

Usually, the first implementation of the DSS is made on a pilot plant to show the benefits and profits of using this kind of tools. Multi-utility companies obviously want to assess the real improvement in management and how the DSS fits their decision-making processes before extending it to a number of plants.

Having identified the pilot plant, the following steps must be taken to set up the DSS for a new customer:

- defining the list of users who can access the DSS. OptiEPM is a multi-user, multi-plant application: this means that the same customer can configure as many plants as needed and appoint different users having different operating permission levels:
  - administrators, who are allowed to manage all users and all plants and can modify all the parameters;
  - plant users, who can operate only on allowed plants and modify a restricted set of parameters.
- Identifying historical series of energy production and weather conditions (temperatures) and revise them to:
  - erase outliers (abnormal peaks overtaking the production capacity of the plant, irregular zeros (in the middle of two regular values), other anomalous signals);
  - verify consistency of correlated measures: sometimes the total production of a group of machines is measured along with the production of the single machines; in this case it cannot be automatically assumed that the sum of the single signal complies with the signal of the total, and it is necessary to define which signals are reliable and which have to be discarded;
  - as to weather series, discard days lacking of at least 4 h of measured temperature.
- Uploading revised historical series of energy production and weather conditions into the database.
- Setting up the plant configuration, defining all the data explained in Sect. 4.3.2

- Setting up all input data flows, identifying available significant measures and their tags in the metering system.
- (Optional) Setting up automatic procedures for data cleaning, after having identified specific rules. This step can be skipped since the same data cleaning can be done “by hand” in the system through a dedicated interface.

Since OptiEPM is delivered as software as a service, users access the application through a browser and then use web interfaces that are basically of two types:

- **tables**, both for parameter configuration and for showing numerical results of scenarios;
- **graphs**, for an immediate representation of numerical results.

In order to configure the plant, the user is provided with a set of tabs to design the plant and set technical and economic parameters. The first tab is an Editor where the user can upload a plant scheme as background and then add single machines by selecting them from a menu and adding them to the scheme by drag-and-drop. The other tabs show a table of parameters (both technical and economic) and their respective values, editable by the user (Fig. 4.4 shows an example).

A similar interface is provided for single machines, where the user can set specific parameters such as production performance, yearly constraints, and maintenance costs. After setting all the parameters, the user can launch the yearly budget. The result is shown in Fig. 4.5: the lines are the forecasted heat demand, cool demand and electricity demand (for an energy service contract). The histograms under the lines show the production of the respective energy; also, for heat, the possible dissipation (the part of the histogram above the line of demand), for electricity the possibility of selling to or buying from the market.

The yearly budget is the starting point for shorter interval optimizations. The user interface for the daily operations is very similar, and is shown in Fig. 4.6:

Impianto DEMO

EDITOR | COMPOSIZIONE | PARAM. TECNICI | PARAM. ECONOMICI | ALTRI PARAMETRI

ID	Parametro	Valore
ABILITA_DEPURAZIONE_AUTOCON...	Abilita la depurazione della stima degli autoconsumi dai tot...	true
ACCISE_CIVILI	Accisa gas uso civile (€/mc)	0.2139874
ADD_DAY_GESTIONE_BT	Pianificazione a N giorni	2
ALWAYS_LOAD_CONSUNTIVO	Carica sempre il consuntivo anche es. in caso di ottimizza...	false
CSV_TAG	Identificatore nel file CSV	
CSV_TAG_CALORE_ACQUA	Tag di CALORE_ACQUA nel file CSV	
CSV_TAG_ELETTR	Tag di ELETTR nel file CSV	
CSV_TAG_ELETTR_MW	Indica se la misura elettrica è in MW/MWh invece che kW/...	false
CSV_TAG_ELETTR_MWh	Indica se la misura elettrica è un contatore progressivo	false
CSV_TAG_ELETTR_PRENOTATO	Tag di ELETTR (energy service) nel file CSV	
CSV_TAG_ELETTR_PRENOTATO_MW	Indica se la misura elettr. energy service è in MW/MWh inv...	
CSV_TAG_ELETTR_PRENOTATO_MWh	Indica se la misura elettr. energy service è un contatore p...	
CSV_TAG_FRIGORIE	Tag di FRIGORIE nel file CSV	
CSV_TAG_FRIGORIE_MW	Indica se la misura delle frigorie è in MW/MWh invece che ...	false
CSV_TAG_FRIGORIE_MWh	Indica se la misura delle frigorie è un contatore progressivo	false

Fig. 4.4 Example of a parameters table tab

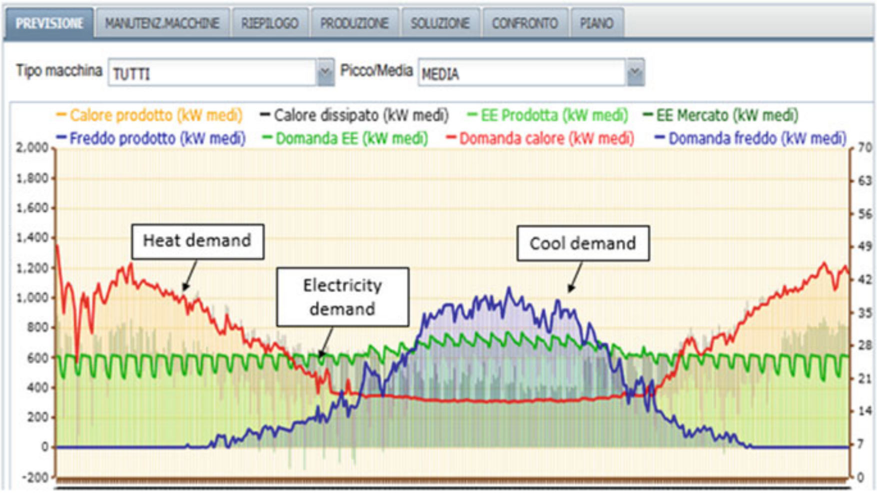


Fig. 4.5 Example of yearly production plan

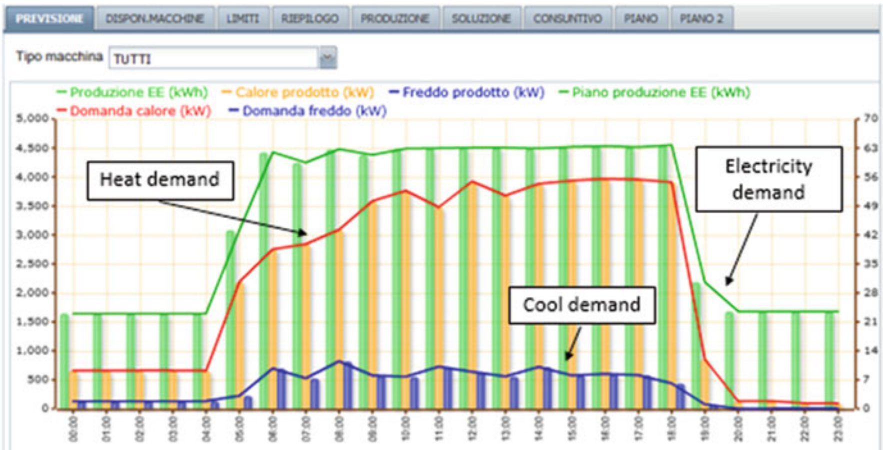


Fig. 4.6 Example of daily production plan

cogenerators are working all day long producing electricity and heat, to fulfill respectively the electricity demand and heat demand. A cooling demand is also present from midnight till 8:00 pm, and consequently electric refrigerators are scheduled to work.

4.3.4 Optimization Module

OptiEPM solution has been developed using a Google Web Toolkit and Enterprise Java Beans 3 framework.

As for the optimization module, a Mixed-Integer Linear Programming (MILP) problem is formulated and solved on a remote server.

Decision variables are associated with the activation level, either discrete or continuous, for each machine in the plant and time period. The main constraints of the model impose the balance of energy flows and the demand for each period and each energy type and the technical feasibility of the production plan, by considering, for example, the maximum and minimum number of turn on/off of a machine per day.

Possible non-linear efficiency curves of the machines are approximated in non-convex piecewise linear functions and linearized in the model. Time is discretized in a finite set of time intervals, having a duration defined by the user. Smaller time windows yield to more detailed production plans but also increase the size of the associated problems and the corresponding computational effort required for their resolution.

MILP problems can be addressed using standard techniques (for example the branch and bound approach) implemented in various commercial software packages. Anyway, in some cases, the duration of the time intervals together with the number of machines in the plant, and complex efficiency curves, make the model intractable if approached "directly", by using commercial software aiming at the optimal solution. Matheuristics (see e.g. [5]) based on time and machines decomposition have therefore been implemented in OptiEPM to reach near optimal solutions in a reasonable amount of time. The key idea of the heuristic strategies consists in the resolution of several easier sub-problems rather than directly addressing the original large MILP formulation. The solutions produced by the heuristic are nearly optimal. As to the quality of the solution, a preliminary testing conducted on small size instances, which can be solved by exact MILP, showed that our metaheuristic produces nearly optimal solutions. On large size instances, the evaluation of the gap, with respect to the optimal solution, is not possible due to the difficulty of solving the instance and the poor quality of the MILP lower bound. However, solutions in different applications have been evaluated by practitioners as reliable and of high quality. For what concerns the execution times, the solution of a medium sized plant and a yearly budget instance goes from roughly half an hour to an hour. Day-ahead planning instances, of minor practical interest, can be solved within 1 min, while a single month requires roughly from few minutes to half an hour. For the most complex plant (11 machines, cooling production, heating production serving two lines—90 and 120 °C—electricity production) the time to solution rise to 2 h for the budget process. Times has been measured on a Intel® Xeon® CPU with two cores of 3.30 GHz and 64 GB of RAM. A deep statistical analysis of computing times and solutions' quality would go beyond the scope of the paper.

4.3.5 System User Experience

OptiEPM has been used for more than 2 years by two big multi-utilities of northern Italy, one for managing six plants connected to the same number of district heating networks, the other for managing an energy service contract for a hospital. The manual management of the operations in this kind of plants is based on the experience of the plant manager that, taking into account the heat demand of the day N-1 and the weather forecasts for the day N, defines the unit commitment for the day N. As for the budgeting process, yearly targets are defined using a number of facts sheets where are set: the plant configuration (especially performance ratios), selling and buying prices and regulatory constraints. All the data regarding plant's operations are stored in a specific database, different from the one used for financial data; regular contacts between plant managers and financial controllers are required for data sharing and revised budgeting. The introduction of OptiEPM has enabled the possibility to manage a "holistic" view of the plant in the same system, simplifying both the daily tasks of the managers and the budgeting process; the user-friendly interface allows an easy access to information and quick changes to parameters when necessary. Also, after these 2 years, the following remarks can be pointed out:

- the DSS has proved its effectiveness, raising profits by about 5–10 % per year; this is a direct consequence of the more precise machines scheduling, compared to manual management that tends to set machines monthly or, at best, weekly;
- the best results are in mid seasons (i.e., spring and autumn), because the probability of unstable weather is higher and manual operations are less responsive to unexpected weather conditions;
- OptiEPM allows the plant manager to perform what-if analyses to determine the best period for ordinary maintenance operations;
- the possibility of performing what-if analyses also allows operators to evaluate how to manage other types of constraints, such as the number of working hours of the engines before extraordinary maintenance operations. For example, one plant manager has decided to limit daily working hours of the engines to avoid the need for an extraordinary maintenance operation before the end of the energy service contract. Using OptiEPM, a comparison between two scenarios has been made: the BAU scenario vs. the no-limit scenario, with no working hours constraint on the engines; the result was that the extra profits gained by making the engines work 24 h a day was higher than the cost of the extraordinary maintenance operation showing that, at least, this decision should be thoroughly considered.

To obtain the best results, the plant manager has to keep the system updated. This implies that the performance production ratios of the machines have to be updated after every ordinary maintenance operation and/or after direct measuring of the performance.

4.4 Conclusions

The great value of these DSSs relies on the acceleration of processes that have been requiring such a long time to be concluded. Manual management based on facts sheets is a time consuming activity; since DSSs extremely simplify the generation and comparison among different scenarios, users can concentrate their efforts on the interpretation of results in a more efficient and effective way; also, more data are provided to draw the final decision (i.e. financial data), that is still a human one. Yet, these DSSs are not to be considered financial controlling tools, since the economic data provided are not as detailed as a financial controlling activity requires. Also, reports generated are limited to prices of materials.

The more operational the problem is, the better: in this case a DSS can generate an optimized production plan ready for the machines. Plant managers and their staff consequently can save some time to spend in other activities. On this point, we should remark that sometimes plant managers are cautious about the solutions proposed by the DSS, because these solutions may not be similar to the manually defined ones and they have to be convinced of the soundness of the new approach. In some cases they ask to modify the DSS to generate solutions more similar to the "traditional" ones. The difficult job is to convey the experience of the manager in the DSS without inheriting his mistakes!

Further research activities in this field should consider the role of stochasticity in energy demands.

References

1. Aringhieri, R., Malucelli, F.: Optimal operations management and network planning of a district heating system with a combined heat and power plant. Ann. Oper. Res. **120**(1–4), 173–199 (2003)
2. Bøhm, B., Pálsson, H., Larsen, H.V., Ravn, H.F.: Equivalent models for district heating systems. In: Proceedings of the 7th International Symposium on District Heating and Cooling. Nordic Energy Research Programme (1999)
3. Bordin, C., Gordini, A., Vigo, D.: An optimization approach for district heating strategic network design. Eur. J. Oper. Res. **252**, 296–307 (2016)
4. Borghetti, A., Frangioni, A., Lacalandra, F., Nucci, C.A.: Lagrangian heuristics based on disaggregated bundle methods for hydrothermal unit commitment. IEEE Trans. Power Syst. **18**(1), 313–323 (2003)
5. Boschetti, M.A., Maniezzo, V., Roffilli, M., Röhler, A.B.: Matheuristics: optimization, simulation and control. In: Hybrid Metaheuristics. Lecture Notes in Computer Science, vol. 5818, pp. 171–177. Springer, Berlin (2009)
6. Carrión, M., Arroyo, J.M.: A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem. IEEE Trans. Power Syst. **21**(3), 1371–1378 (2006)
7. Illerhaus, S., Verstege, J.: Optimal operation of industrial chp-based power systems in liberalized energy markets. In: International Conference on Electric Power Engineering, 1999. PowerTech Budapest 99, p. 210. IEEE, Piscataway (1999)
8. Kazarlis, S.A., Bakirtzis, A., Petridis, V.: A genetic algorithm solution to the unit commitment problem. IEEE Trans. Power Syst. **11**(1), 83–92 (1996)

9. Lahdelma, R., Hakonen, H.: An efficient linear programming algorithm for combined heat and power production. Eur. J. Oper. Res. **148**(1), 141–151 (2003)
10. Larsen, H.V., Bøhm, B., Wigbels, M.: A comparison of aggregated models for simulation and operational optimisation of district heating networks. Energy Convers. Manag. **45**(7), 1119–1139 (2004)
11. Larsen, H.V., Pálsson, H., Bøhm, B., Ravn, H.F.: Aggregated dynamic simulation model of district heating networks. Energy Convers. Manag. **43**(8), 995–1019 (2002)
12. Li, T., Shahidehpour, M.: Price-based unit commitment: a case of lagrangian relaxation versus mixed integer programming. IEEE Trans. Power Syst. **20**(4), 2015–2025 (2005)
13. Loewen, A., Wigbels, M., Althaus, W., Augusiak, A., Renski, A.: Structural simplification of complex dh-networks-part 1. Euroheat Power Fernwarme Int. **30**(5), 42–44 (2001)
14. Loewen, A., Wigbels, M., Althaus, W., Augusiak, A., Renski, A.: Structural simplification of complex dh-networks-part 2. Euroheat Power-Fernwarme Int. **30**(6), 46–51 (2001)
15. Mantawy, A., Abdel-Magid, Y.L., Selim, S.Z.: Unit commitment by tabu search. IEE Proceedings C - Generation, Transmission and Distribution, vol. 145, pp. 56–64. IET (1998)
16. Ostrowski, J., Anjos, M.F., Vannelli, A.: Tight mixed integer linear programming formulations for the unit commitment problem. IEEE Trans. Power Syst. **1**(27), 39–46 (2012)
17. Padhy, N.P.: Unit commitment-a bibliographical survey. IEEE Trans. Power Syst. **19**(2), 1196–1205 (2004)
18. Park, Y.: State of the art report of Denmark, Germany and Finland. Simple models for operational optimization. Department of Mechanical Engineering, Technical University of Denmark (2000)
19. Rong, A., Hakonen, H., Lahdelma, R.: A dynamic regrouping based sequential dynamic programming algorithm for unit commitment of combined heat and power systems. Energy Convers. Manag. **50**(4), 1108–1115 (2009)
20. Song, Y., Chou, C., Stonham, T.: Combined heat and power economic dispatch by improved ant colony search algorithm. Electr. Power Syst. Res. **52**(2), 115–121 (1999)
21. Thorin, E., Brand, H., Weber, C.: Long-term optimization of cogeneration systems in a competitive market environment. Appl. Energy **81**(2), 152–169 (2005)
22. Wood, A.J., Wollenberg, B.F.: Power Generation, Operation, and Control. Wiley, New York (2012)
23. Zhao, H.: Analysis, Modelling and operational optimazation of district heating systems. Centre for District Heating Technology, Technical University of Denmark (1995)
24. Zhao, H., Holst, J.: Study on a network aggregation model in dh systems. Euroheat Power **27**(4–5), 38–44 (1998)

Chapter 5
Birth and Evolution of a Decision Support System in the Textile Manufacturing Field

Pierpaolo Caricato, Doriana Gianfreda, and Antonio Grieco

Abstract We present the evolution of a Decision Support System that was developed for a company that produces high-tech fabrics. The project started in 2010 within a cooperation agreement between a public Italian university and the firm, initially addressing what was perceived as the main and more peculiar aspect of the decision process related with the production planning: namely the sheet booking process. We designed and developed a DSS that implemented a mathematical programming algorithm based on combinatorial optimization to solve the very peculiar variant of the cutting stock problem that could be used to model the decision process. The results of the usage of the DSS outperformed any estimation the company had expected from the project, leading not only to the automation of the decision process, but also to a large enhancement in the material usage rates. We present a short outline of the improvements achieved with this first tool. The positive results obtained with the first DSS led to two further evolutions of the tool: the former was developed 2 years later, while the latter is currently being developed.

5.1 Introduction

As reported on the official web site http://www.saati.com/company.php

> SAATI is an Italian multinational company that develops, produces and commercializes highly advanced technical precision fabrics and chemicals for industrial use since 1935.

Its textile production covers the whole process that starts with the looms weaving raw fabric reels and ends with the realization of semi-finished items used in printing, chemicals and filtration or with the realization of finished fabric reels used by industrial customers in their production processes.

In this chapter we present the results of a collaboration between SAATI and the public University of Salento in Italy, that began in 2010 and is still active today. The collaboration started with the study and realization of a Decision Support System

P. Caricato (✉) • D. Gianfreda • A. Grieco
DII – Università del Salento, Via Monteroni, Lecce, Italy
e-mail: pierpaolo.caricato@unisalento.it

© Springer International Publishing Switzerland 2016

J. Papathanasiou et al. (eds.), *Real-World Decision Support Systems*,
Integrated Series in Information Systems 37,
DOI 10.1007/978-3-319-43916-7_5

for the needs of a very specific decision area in the production planning process and subsequently evolved with tools embracing a wider and wider area of the production related decisions taken in the firm.

5.1.1 The Problem

The main products that the firm sells are finished fabric sheets. The production process includes the weaving phase, one or more physical and chemical treatments on the raw fabric and finally a cutting phase, that produces smaller sheets of finished fabric from larger ones.

The problem that was felt by the company's production managers as the most peculiar—and hence strategic for the success of a computer-driven decision support project—was the so called *apparto* process. *Apparto* can be roughly translated as *booking*: the user (the production planner) needs to decide how to use the available finished or semi-finished fabric reels to fulfill the customer orders.

The result of a booking process is the *reservation* of smaller areas of a larger fabric reel for the needs of specific orders. The actual production of the final fabric sheets does not necessarily take place when the reservation is done. On the contrary, in most cases the booking process covers an entire reel through several iterations, as long as new customer orders fill the orders' portfolio, and the actual cutting only takes place when the due date for one or more of the orders allocated to the reel gets closer.

5.1.2 The Need for a DSS

The allocation of orders to available fabric reels is a very complex process that must take into account several aspects, among which:

- each fabric reel is characterized by a *quality map* and the orders require specific quality levels;
- orders are expressed in terms of an overall required quantity (a very large rectangle that can have one or both dimension specified in terms of a desired value and an acceptable deviation from it), but they must be fulfilled in terms of smaller sheets (that share the same height of the overall order, but different length requirements);
- the available fabric reels can be:
  - large reels of not yet finished fabric: the quality map of such reels can only be estimated and their availability for production can be only roughly estimated according to standard lead times;

– large reels of semi-finished fabric: a detailed quality map and a reliable availability are given for such reels, but they often already have reservations descending from previous booking processes;
– small reels or sheets of finished fabric: these items are immediately available for cutting as they are cutting remainders of not entirely used fabric reels.

A more comprehensive presentation of these aspects is treated in Sect. 5.5.

The amount of detailed data to be considered both for the orders and for the available fabric reels, as well as the number of aspects to be pondered made the need for a DSS very clear to the firm managers.

5.1.3 Target Group

The booking activity was traditionally managed by a group of employees, who were full-time dedicated to this task. A supervisor coordinated their work, providing the employees with strategic drivers that should guide them as much as possible in their decisions.

The booking activity was conducted on an SAP custom user interface that allowed an employee to:

- select a fabric reel within a tabular list including all the available ones, with a subset of reels features that were useful to the user (dimensions, tabular list of quality zones within the reel);
- see a list of orders in the portfolio with a height requirement compatible with the height of the selected reel;
- select one of the orders and one of the quality zones of the selected reel to allocate the order to the zone.

The supervisor was the main figure involved in the analysis of the objectives and constraints that the DSS had to consider. The employees where also deeply involved both in the analysis phase and throughout the development of the system as key users.

5.1.4 Existing Procedures

The solution adopted in the firm prior to the development of the DSS, the SAP based procedure described before, had several drawbacks, mainly:

- the results achievable by the user depended on the initial reel selection;
- evaluating different booking possibilities selecting different initial reels and possibly different orders to be allocated was extremely time-consuming;
- the long and *manual* activity required for the process led the user to almost always prefer large reels, where more orders could be allocated;

- as a consequence, smaller but already available finished reels were rarely used, leading to a worse customer response and to an always growing finished reels warehouse;
- the lack of a graphical representation of the reels and of the orders reservations required the user to annotate different possible solutions outside of the system (often on paper), evaluating them using a calculator or a spreadsheet and possibly making non-traced errors.

The main drawback of the former tool was that it had no embedded intelligence, leaving the user completely free (and completely alone) to decide which reels to examine and how to use them. It was just an interactive way to select a reel, place bookings for one or more orders on it, and then bring the data about the user's choice back into SAP.

The UI of the former tool was not designed to help the user take decisions: it was rather meant as a way to input decision data into SAP, considering the decision process as something that had to happen outside SAP. Consequently, the only *help* to the user was the automatic calculation of the unused reel space, and it was just a *collateral effect* due to the need for SAP to be able to successfully close a transaction on a reel only when all its available capacity had been used, while no other decision support, such as the possibility to save two or more alternative usage scenarios for a same reel, was provided.

In this context, the quality of the achieved solution was largely related to the time the user had to accomplish its task: the more time he had, the higher was the possibility for him to study more reels, jot down different possibilities on paper or on a spreadsheet file, and then decide for the best one among them. The approach was in no way exhaustive, since only some of the available reels could be examined, and only considered a *local* objective (i.e. saturate the usage of the selected reel), losing the possibility to consider the overall materials usage, not to mention the possibility to also consider further objectives.

5.1.5 Classification

According to Power's [15] and [16] expanded DSS framework, the DSS case study we present fits perfectly within the model-driven DSS type.

- the *dominant DSS component* is an optimization model based on integer linear programming, which acts as the *enabling technology*;
- the *target users* are the booking process supervisor and the employees he coordinates;
- the *purpose* of the DSS is to allow both an automation and an optimization of a relevant part of the planning and scheduling process;
- the deployment technology is a stand-alone PC, integrated with the main information system through data exchange procedures.

5.1.6 *Underlying Technologies*

The dominant technology component or driver of the presented DSS is a quantitative model of the problem to be addressed. Specifically it uses an optimization algorithm based on ILP (Integer Linear Programming) and lets the user compare different scenarios obtained varying configuration parameters. Hence, accordingly to [15] and [17], it perfectly fits into the definition of a model-based DSS.

Hence, the main technologies that underlie the success of the DSS in the firm, are:

- combinatorial optimization techniques (mainly pattern generation and integer linear programming), that allow the DSS to obtain excellent solutions even if the number of all possible solutions is typically too large for an exhaustive analysis;
- the usage of a user friendly UI (User Interface), that allows the DSS user to define different scenarios, characterized by specific constraints that have to be respected or by parameters' values corresponding to the different strategies to be pursued.

From a more Information Technology point of view, the underlying technologies that the DSS uses are:

- Microsoft Visual Studio 2013 is the tool that was used to develop the DSS: in particular, C# was the language adopted to implement both the UI and the algorithms required to solve the decision problems;
- IBM ILOG CPLEX 12.6 is the mathematical programming library that was used to solve the instances of ILP problems that are created by the decision algorithm.

5.2 System Requirements Analysis

The system requirements analysis was conducted with an iterative approach, refining the requirements through intermediate releases of the tool, using the following scheme:

- interviews and meetings with the target group were used to define an initial set of requirements;
- meetings with the information systems staff were held to define what data was already available in the previously used booking system and what had to be defined and formalized;
- the architecture to enable an effective interaction between the DSS and SAP was defined;
- several versions of the tool were released, starting with minimum initial requirements and then enhancing the required features, the DSS decision model and the data and interaction model;
- a *side-by-side* usage of the previous procedure and of the DSS was conducted for a month, using it to gather final fine tuning requirements.

5.2.1 Requirements Gathering

As reported in Sect. 5.1.3, two main types of figures were involved in the requirements gathering: the booking supervisor and the booking employees.

The contribution from the former figure was expected in the definition of standards that the DSS should respect or at least try to achieve: hence the supervisor was mainly involved in the definition of the objectives the DSS had to pursue as well as in the definition of the KPIs (Key Performance Indicators) to be calculated to monitor the performances of the DSS.

The main objective of the booking DSS was the usage of the available fabric reels. The survey with the supervisor helped define the following important classification. When a fabric reel is used by the booking process, its overall area is divided into smaller areas that can be referred to as:

used reserved for an order,
unusable unused due to quality issues (this area cannot be used by any order),
reusable an area that is large enough to be stocked for future usage,
trim loss an unused area that is not large enough to be reused.

Such classification allowed the introduction of the main KPI for the booking process, i.e. the rate between the trim loss and the available area (not including the unusable part). As part of the requirements gathering, a survey was also conducted to obtain a set of historical data about the defined KPI.

Further KPIs were also defined, to capture other aspects of the booking process. In particular, the average size of the used reels and the number of reels that are completely used by the process emerged as two more KPIs to be considered.

The requirements gathering also took advantage of the employees' experience to acquire from them:

- general impressions on the previously used booking system;
- general expectations they had from the DSS project;
- guidelines they followed during the booking process.

As reported in Sect. 5.1.4, the main users' criticism about the former tool was related to its usability, since its quantitative performance was, indeed, entirely in the hands of the user, without much help from the tool. This judgement on the existing procedures made it clear how important it was to design a system that was not only efficient in terms of its functionality, but also, at a same level of importance, effective in terms of its usability and, more in general, in terms of non-functional requirements, as defined in [2] and [3]. The work of Parker [13] was also of great help in being focused on defining the *enquiries* inherent the decision process.

Finally, the information system architecture was also part of the requirements gathering, to define with the IT staff of the company the most effective way to have the DSS *connected* with the rest of the system.

5.2.2 Requirement Determination and Definition

The definition of the KPIs described before was the main quantitative aspect that was defined. Other detailed aspects that were defined were the ones related to the IT architecture: the company's requirement was to have the DSS behave as a plug-in tool, intending that it had to be able to receive data from SAP, conduct its user-guided processing and transmit its user-validated results to SAP, but without substituting the existing process.

Indeed, the DSS was required to be a stand-alone Windows application that received data in the form of well-formatted text files and exported its results in the same form. The company's IT staff would implement SAP-based procedures to prepare the data required by the DSS as well as to develop the code needed to import back to SAP the results of the processing exactly as if it had manually been produced by one of the employees working with the previous procedure.

Through this approach, the DSS fulfilled the prerequisites to be used in a wide set of ways, ranging from an almost entirely automatic processing on all orders and available reels, to being used only on specific subsets of data as well as being completely bypassed.

Finally, a general sense of un-satisfaction with the SAP user interface emerged, though it did not translated into an explicit requirement.

5.2.3 Final System Proposal

The result of the requirements gathering was a first outline of the DSS focused on the following aspects: its tasks, its objectives and its IT architecture.

The DSS general task was to support the booking employees in their daily activity, providing them with a way to set some strategic parameters and automatically obtain the best possible booking according to the imposed parameters. A fine tuning of these parameters would be carried out within the period of side-by-side usage, in order to have such parameters as the weight of each KPI during the booking process fixed to reasonable values that would be actually used in production afterwards.

The DSS objectives were defined consequently with the KPIs defined with the supervisor, hence removing much of the arbitrariness of the previous booking procedure, where the actual pursued objectives could vary with each employee.

The main aspect regarding the IT architecture was the need to have the DSS as a *parallel/offline* system, that could be (at least initially) turned on or off, or used only on subsets of the orders and available material, without needing any IT staff action.

5.3 System Design and Development

The DSS works as an independent system which provides an integration with the management information system only limited to data exchange. This choice was required by the company and justified by the objective to keep the *intelligence* of the decision support system strictly separated from the rest of the information system, in order to be able to easily and independently modify the two tools: the *custom-tailored* DSS on one side and the *ordinary/standard* orders management tool on the other.

As a result of this choice, the Booking DSS is a stand-alone Windows application, supported by a local database where inputs are loaded and final results are saved. On its turn, the external information system (SAP) has its own database, where both historical and incoming data are collected. Data exchange between the external information system and the DSS occurs through plain text files, characterized by well defined structures.

The user selects a set of data in the information system and exports the required data into text files the DSS is able to import into its local database. The DSS UI exposes the selected data to the user and allows him to set parameters, define custom constraints and check the data for coherence and consistency. The user can then fix possible issues or start with the processing if all checks are passed.

The final results are automatically stored in the local database and shown to the user, who can partially or totally accept them or decide to completely reject the solution (possibly saving it as a reference for further scenario analysis), in order to execute another run with different parameters. When the user is done with scenario analysis, the results of the scenario selected by the user are exported into text files, that are imported by the information system and stored within SAP data structures exactly as if they had been achieved using the previous booking system.

In the actual usage of the DSS, it is also important to define both orders and available material states, to be able to *freeze* some orders or some bookings that took place in previous runs of the DSS (or that were manually decided outside the DSS). For instance, if booking an order onto a specific fabric reel implies moving it from a location to another then, once this movement has taken place, this booking will have to be confirmed in the following runs. On the other hand, other bookings may be easily rediscussed in the following runs, if they can lead to better KPI values.

To summarize, in this DSS case study, a typical lifecycle of a single usage of the DSS consists of:

- importing input data (orders, fabric sheets, weaving forecasts) from text files generated by the information system (possible anomalies are pointed out);
- setting one or more scenarios, i.e. setting user-defined parameters (interval of due dates of orders to be scheduled, subset of fabric sheets and weaving forecasts to be included or excluded, objectives weights, etc.);
- checking input data for coherence and consistency, helped by alerts and views to identify and correct any anomaly;

- automated and optimized booking process, which represents the core decision engine of the DSS;
- evaluating and choosing the preferred solution among the results from the different defined scenarios;
- exporting output data (scheduling of orders and updated maps of fabric sheets) into text files to be sent back to the information system.

5.3.1 The Problem

Each order from a specific customer is typically characterized by a required raw material and a minimum accepted quality degree, along with required height, total quantity to be produced and size of each item (cut length), all defined in terms of a desired value and an accepted tolerance.

Customer orders can be satisfied *booking* rectangular areas of fabric on reels of raw material, on reels of semi-finished fabric or on sheets of residual finished fabric. Each reel or sheet of input material will be available to be cut at different times: finished sheets are immediately available, semi-finished fabric is available within a given date, while raw materials become available after a roughly estimated lead time. Consequently, booking part of an input material for an order allows to obtain an estimation of a date by which that part of the order will be ready to be shipped to the customer.

The general booking task consists in fulfilling the orders with the objective to minimize the trim loss. Actually, the subsequent evolution of the original DSS that is currently used in the firm uses the booking process not only to help the user decide how to use the available materials, but also to estimate the shipment dates for the served orders, according to the expected availability date of the sheets that are used to serve them.

The current DSS, hence, considers two different types of objective. The former, more related to customer satisfaction, is a cost function that depends on the total amount of orders that are expected to be shipped late, weighted by aspects such as the order size, the number of delayed days and the importance of the customer. The latter objective, instead, is more production oriented and uses the trim loss as its main component.

Terminology

In order to be able to explain the data structure that underlies the DSS, a brief though detailed insight on problem-specific terminology needs to be done. A glossary is provided in Table 5.1.

The set of available fabric reels and sheets includes both real sheets and weaving forecasts. Each fabric sheet, located at a certain warehouse, is defined by three maps, describing the evolution of its structure and usage throughout the booking process.

Table 5.1 Glossary

Term	Definition
Order	Is a customer's requirement for a specific finished material, characterized both by its geometric specifications, required quantity and due date
Reel	Is the form in which the raw material (fabric) is made available for the cutting process
Sheet	Is the form in which reusable remainders of previous cutting processes are available for further cutting
Weaving forecast	Is a prevision for a reel that will be available at a certain date
Stock-piece	Is a rectangular zone of a fabric reel or sheet, characterized by an homogeneous quality level and being either free or entirely assigned to a specific order
Map	Is the detailed information about the stock-pieces that are contained in a reel or a sheet
Booking	Is the allocation of customer orders to fabric reels and sheets that hence produces the allocation maps for the used materials

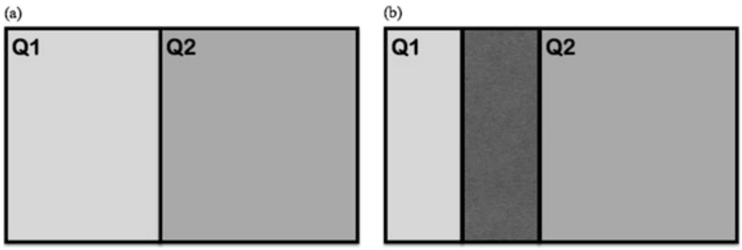


Fig. 5.1 Initial map. (**a**) entirely available map (**b**) partially occupied map

On its turn, each map is composed by a set of stock-pieces, which are portions of fabric, each characterized by a specific size, quality level and position within the reel. The first map is called *initial map* since it denotes the initial composition of the fabric sheet, before the booking process; in other words, it represents the structure of the fabric sheet as it comes from the input files. In most cases, stock-pieces of the initial map are entirely available (Fig. 5.1a), but sometimes they are already booked for specific orders (dark central rectangle in Fig. 5.1b).

The other two maps, called *intermediate* and *final*, represent the further evolutions of the initial map and are respectively referred to an intermediate stage and to the final solution of the booking process. Given an initial map of a fabric sheet, in the intermediate stage of booking process, its stock-pieces are modified, in terms of number and size.

As shown in Fig. 5.2a, the initial map is composed by two stock-pieces, having quality level respectively equal to B1 and B2, where B1 is better than B2.

In the intermediate stage of the booking process, stock-piece 1 is booked for orders O1 (which partially uses the height of the stock-piece) and O2 (which uses

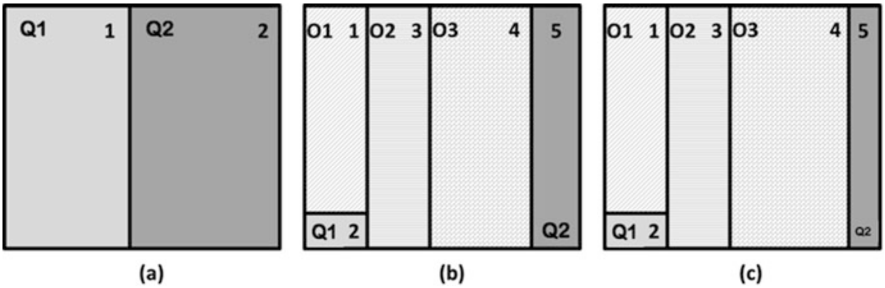


Fig. 5.2 Map evolution. (**a**) initial map (**b**) intermediate map (**c**) final map

the entire height), thus generating three stock-pieces in the intermediate map (two booked stock-pieces and a residual one), as represented in Fig. 5.2b. Stock-piece 2 of the initial map is booked for order O3, which uses the entire height of the map, but not the entire length, thus generating two stock-pieces in the intermediate map (one booked stock-piece and one residual).

After further post-optimizations, the intermediate map can evolve into the final map, where the size of some stock-pieces and the length assigned to orders can change as a consequence of the decrease or even the removal of residuals. As shown in Fig. 5.2c, the length of stock-piece 4 in the final map is greater than that of the intermediate map, while the residual (stock-piece 5) is lower.

Each stock-piece generated in the intermediate or final map always maintains a reference to its generating stock-piece in the initial map. In the example of Fig. 5.2, stock-pieces 1, 2 and 3 of the intermediate or final map are referred to generating stock-piece 1 of the initial map, while stock-pieces 4 and 5 are referred to generating stock-piece 2.

The fabric sheets deriving from weaving forecasts do not have an actual initial map, so an expected realistic map is considered; the same happens for a subset of real fabric sheets, whose real initial map is yet unknown due to quality check issues.

The assignment of orders to fabric sheets is limited by different conditions; the most of them are imposed by customers and derives from the characterization of the orders, while the others are defined by the DSS user. The last group of conditions, called user-defined constraints, forces or forbids the assignment of a set of orders to a specific set of fabric sheets and/or vice versa. Therefore each user-defined constraint is characterized by the sets of orders and fabric sheets which are involved and the category (obligation or prohibition); moreover, in case of obligation, also a *direction* needs to be defined, in order to determine whether the condition starts from orders, from fabric sheets or from both orders and fabric sheets; prohibition is always symmetric.

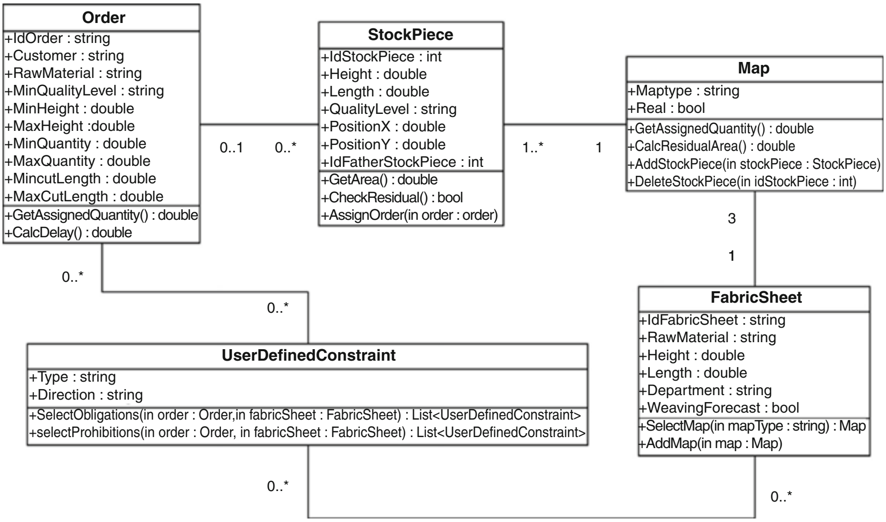


Fig. 5.3 Class diagram

5.3.2 Class Diagram

A static view of the building blocks of the Booking DSS is depicted in the class diagram in Fig. 5.3.

In the class diagram, five objects are modeled, namely: order, fabric sheet, map, stock piece and user-defined constraint.

Order

The Order class represents a single customer order defined by a collection of attributes which identify the request (IdOrder), the customer which submits it (Customer) and all its requirements, in terms of required raw material (RawMaterial), minimum accepted quality degree (MinQualityLevel), minimum and maximum total quantity to be produced (MinQuantity and MaxQuantity), minimum and maximum required height (MinHeight and MaxHeight), minimum and maximum size of each item (MinCutLength and MaxCutLength). Total fulfilled quantity of the order is returned by the GetAssignedQuantity method, while the quantification of the associated delay is calculated by the CalcDelay method.

StockPiece

The StockPiece class denotes a single portion belonging to a map of a fabric sheet and is characterized by geometrical attributes concerning the size of the stock piece (Height and Length) and its coordinates starting from the top-left corner of the map (PositionX and PositionY).

Moreover, each object has a specific quality degree (QualityLevel) and a reference to its generating stock piece in the initial map (IdgeneratingStockPiece). The area of a stock piece, useful for defining the objective function which minimizes the trim loss, is calculated by the GetArea method; then, in order to verify if the portion is a re-usable residual, CheckResidual method is given. From the point of view of orders, AssignOrder method allows to allocate the order specified by order parameter onto the stock piece.

Order and StockPiece classes are interrelated to each other through a logical association, where the cardinality of a class in relation to the other one is depicted. Each object of StockPiece class must be linked to at most one object of Order class, while there is no upper limit to the number of stock pieces associated to each order.

Map

Map class is the representation of the initial, intermediate or final structure (MapType) of a fabric sheet. A Boolean attribute (Real) specifies if the map is real or referred to an expected realistic structure. In order to evaluate the objective function which minimizes the trim loss, GetAssignedQuantity and CalcResidualArea methods respectively return total booked quantity and total area of unassigned stock pieces. When a map is created, its composition in term of stock pieces must be detailed; for this purpose, AddStockPiece method allows to add a new portion specified by stockPiece parameter. On the other hand, in the transition between intermediate and final map, the removal of a stock piece is sometimes required; this can be performed by DeleteStockPiece method.

FabricSheet

FabricSheet class denotes a single fabric sheet with its physical, geometrical and logistical features. Physical properties of a fabric sheet mainly depend on the material (RawMaterial) it is composed by. On the basis of its nature, a fabric sheet is real or forecasted (WeavingForecast). In addition, the identifier (IdFabricSheet) provides further information about the state of the sheet, which can be classified as finished, semi-finished or raw. From a geometrical point of view, a fabric sheet is characterized by its total height (Height) and length (Length). The last category of features concerns the location of the fabric sheet (Department), often involved in the limitations imposed by user-defined constraints. Since the evolution of the structure of each fabric sheet is represented by three maps, SelectMap method allows to choose a single map, whose type is specified by mapType parameter, which can assume Initial, Intermediate or Final values. Instead AddMap method introduces a new map, defined by map parameter.

One map may include multiple stock pieces, while one stock piece may belong to only one map. On its turn, a fabric sheet has at most three maps, while one map is referred to only one fabric sheet.

UserDefinedConstraint

Finally, UserDefinedConstraint class represents a single user-defined constraint, whose behavior is defined by a couple of attributes, which state if the condition is an obligation or a prohibition (Type) and, only in case of obligation, if the limitation starts from orders, from fabric sheets or from both orders and fabric sheets (Direction). SelectObligations and SelectProhibitions methods allow to select respectively the set of obligations and of prohibitions referred to the order and the fabric sheet specified by order and fabricSheet parameters.

Concerning the logical connections between Order and UserDefinedConstraint classes, the relationship is *many-to-many*, since one order may be involved in multiple constraints, while a user-defined constraint may include zero to many orders; the same is for the association between FabricSheet and UserDefinedConstraint classes.

5.3.3 Activity Diagram

In the DSS case study, three main actors are involved, namely: the company information system, the DSS user and the DSS software. The activity diagram reported in Fig. 5.4 displays the sequence of activities and identifies the actors responsible for each action.

The activities start with the external information system, which generates input files by collecting orders, available and programmed fabric sheets belonging to a frozen time window (Generate input files action).

When the text files are available, the DSS user sets their paths and then launches loading operations (Set the paths of input files and Launch loading operations actions).

The DSS software tries to import input data into its local database (Load input data action) and, if some anomalies occur, returns a list of issues (Display anomalies action) to be analyzed by DSS user (Collect anomalies action), in order to be solved by information system (Correct anomalies), which generates new input files to be imported. Instead, if data loading is successfully performed, DSS user can set the parameters of the problem and define user-constraints (Set parameters and Define user-constraints actions).

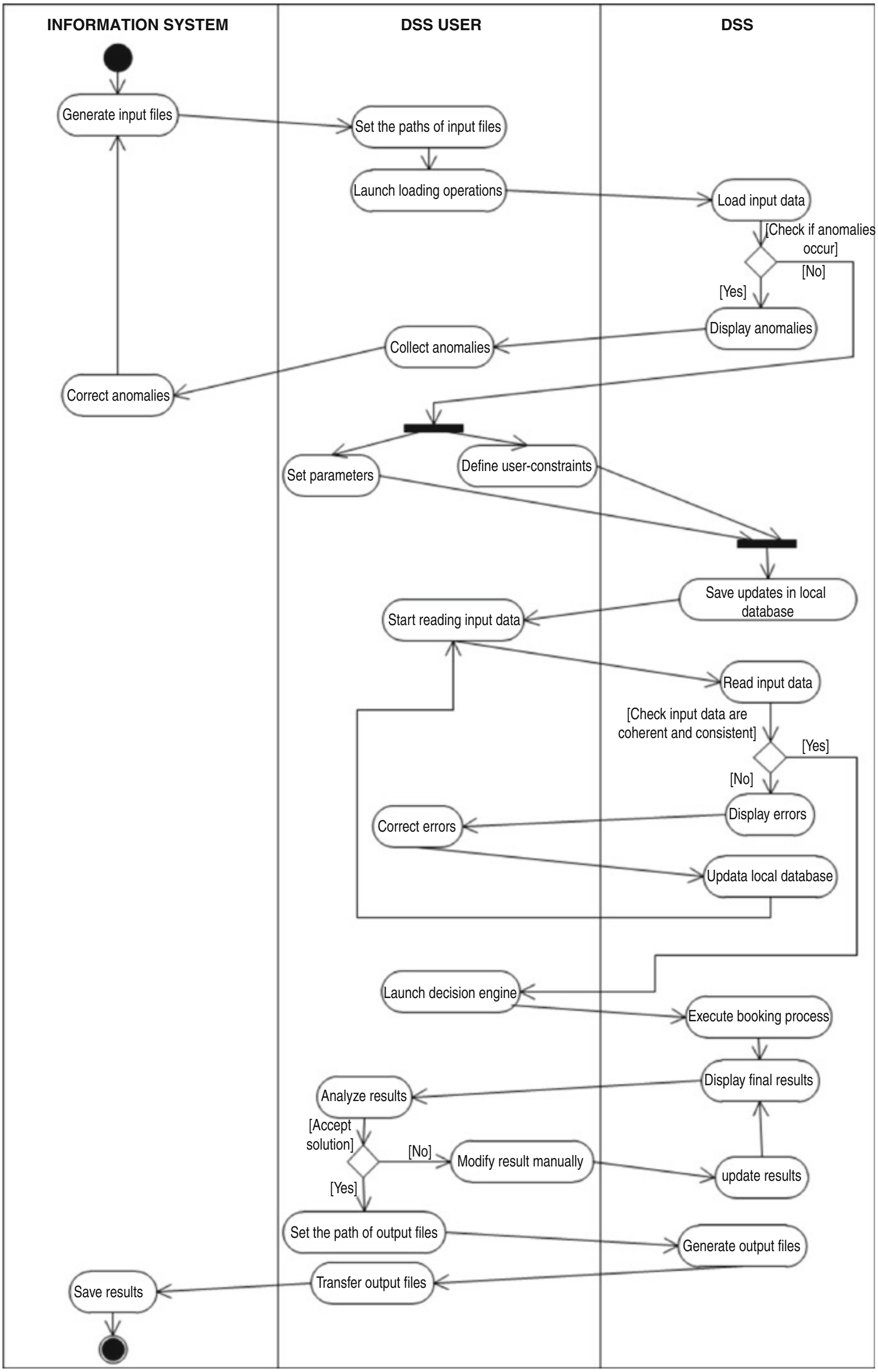


Fig. 5.4 Activity diagram

To consider these settings in the execution of the booking process, DSS software stores the values of general parameters, the interval of due dates of orders to be scheduled, the subsets of fabric sheets and weaving forecasts to be included or excluded and the user-defined constraints into its local database (Save updates in local database action).

After DSS user launches reading operations (Start reading input data action), DSS software reads input data from input tables of the local database (Read input data action) and then checks them for coherence and consistency. If some problems occur, a list of errors is shown (Display errors action) to DSS user, who can analyze and manually correct any anomalies (Correct errors action). Each change applied to input data or parameter settings is saved into the local database (Update local database action), which now must be read again.

If the set of input data is coherent and consistent, DSS user is enabled to launch the decision engine of Booking DSS (Launch decision engine action). After executing the booking process (Execute booking process action), DSS software shows final results (Display final results action), by providing a set of indicators which allows DSS user to perform both a punctual and global evaluation of the solution (Analyze results action). DSS user is now able to decide to accept entire solution or a part of it. Indeed if the allocation of a fabric sheet is not satisfying from the point of view of one or more KPI indicators, DSS user can manually excluded it (Modify results manually action); this operation obviously generates an update in the output tables of the local database.

After possible manually changes, when the solution displayed by interface is entirely acceptable, DSS user sets the paths of output files (Set the paths of output files action), where results concerning the schedules of the orders and the final configurations of the fabric sheets must be saved, in order to allow DSS software to produce output files (Generate output files action).

The last activity of DSS user consists of moving output files to information system (Transfer output files action), which can finally update its own database (Save results action).

5.4 User Interface

The Booking DSS provides a graphical user interface that allows the user to manage each phase of the booking process with the objective to always be responsive and to help the user not only through traditional tabular displays of the data, but also through a graphical representation of the data that was specifically designed for this industrial case.

5.4.1 A Typical DSS Usage

Import from SAP

Before starting using the DSS, the user needs to generate, on SAP, all the input files required by the booking process. When these files are available, in the paths setup on the DSS UI, the input data are imported into the local database.

Each text file is characterized by a pre-defined structure and contains a specific set of input data, such as orders to be produced with their initial schedules, available fabric sheets with their maps, weaving forecasts, etc. Once finished with the loading operations, an alert reports the successful outcome or points out any anomalies encountered which are then displayed to the user, so that he can analyze the issues and solve them outside the DSS (typically on SAP or contacting the IT staff for major problems).

Parameters Setting

Import operations are followed by a parameters setting phase, which may also include the entry of user-defined parameters in the DSS. While import from SAP is automatic, parameters setting requires more interaction. The parameters to be set are:

- interval of due dates of orders to be scheduled (since only orders having due dates included in the fixed interval are considered), which is set using two date/time pickers;
- subset of fabric sheets and weaving forecasts to be considered, selected on the basis of:
  - raw material, by adding the materials to be included or excluded in specific text fields;
  - type of map, by modifying the state of a check box, in order to enable or disable the fabric sheets without a real map;
  - type of fabric sheets, by changing the state of a check box which enables or disables weaving forecasts;
- general parameters, partitioned into different groups depending on the phase of booking process they are involved (configuration, run, model, patterns generation, post-optimizations); these parameters and their descriptions are displayed with the default values, which can be modified by interface.

User-Defined Constraints

Together with user-defined parameters, a set of conditions to be applied in booking process, called user-defined constraints, can be imposed. The user can define new

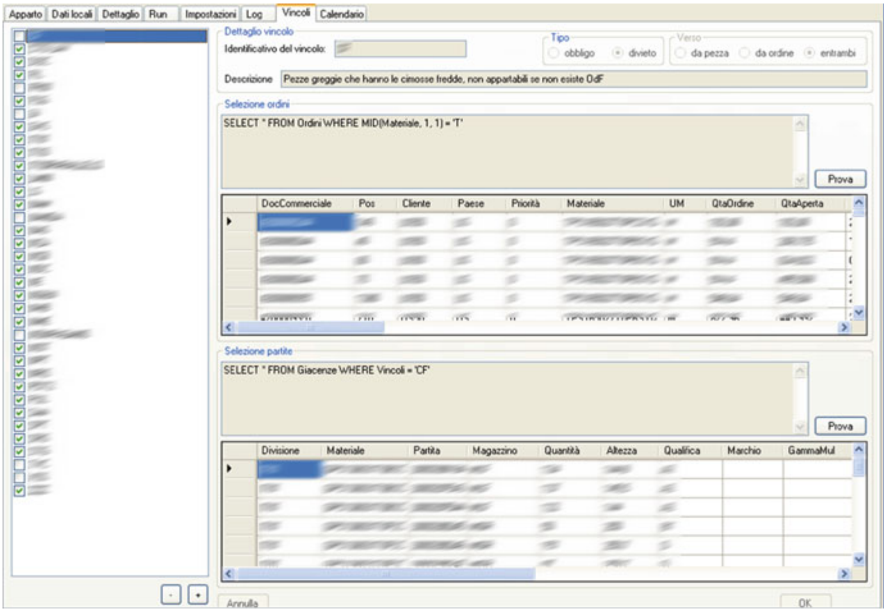


Fig. 5.5 User-defined constraints

conditions and remove or update existing constraints. Each change concerning user-defined constraints is automatically saved in a specific table of the local database. As represented in Fig. 5.5, the set of user-defined constraints is displayed in a check box list, where the user can enable or disable one or more conditions. The selection of an element of the list shows all features of the constraint:

- name and description;
- type, in order to classify the constraint into the category of obligation or prohibition;
- direction, specified only for obligation, which states if obligation starts from fabric sheets, from orders or from both fabric sheets and orders;
- selection query of the orders involved in the constraint;
- list of the order selected by the query;
- selection query of the fabric sheets involved in the constraint;
- list of the fabric sheets selected by the query.

All text fields are editable so the features of an existing constraint can be simply modify and then saved in the local database (button "OK"). Concerning selection queries, once modified, the user can try them (button "Prova"), in order to check the selected sets of orders and fabric sheets. An existing user-defined constraint can also be selected to be removed from the check list and therefore from the local database (button "-"). On the other hand, the user can define a new constraint (button "+")

with its main features and then decide to save it in the local database (button “OK”) or to cancel it (button “Annulla”).

Check and Run

After being entirely populated, the local database can be read, in order to verify the coherence and the consistency of input data. One more time, an alert reports the successful or failed outcome of reading and checking operations; all occurred anomalies are displayed by interface, so the user can apply some corrective actions to solve these problems by modifying input data manually, but after that coherence and consistency check has to be performed again.

The launch of booking process follows reading and checking operations. The processing time required to obtain the final solution is in the range of minutes, gradually increasing with the size of the sets of orders and fabric sheets.

Final results and information about the successful or failed outcome of booking process are shown divided by material. Through a combo box, the user can select a single material and evaluate the solution from different points of view (orders, fabric sheets, delay and KPI).

5.4.2 Results and Graphical Maps

From the fabric sheet point of view, the entire set is displayed by default in a list box (see Fig. 5.6), where two different filters can be applied: by booking state and by type of fabric sheet. The first one selects fabric sheets which are totally or partially booked or not booked or incorrect, while last filter allows to include or exclude fabric sheets without map and/or weaving forecasts.

When the user selects a single element of the list box (i.e. a fabric sheet), the header information (e.g. height in centimeters and length in meters) of fabric sheets are shown, together with the graphical representation of the maps associated with the fabric sheet. The user can browse through the available maps using a track bar: when the track bar is at its leftmost value, the initial map is shown, while the final map is displayed when the track bar is at its rightmost value. Initial map represents the stock-pieces of the fabric sheet before booking process, so in general it is entirely available or at most partially booked; instead final map represents the stock-pieces after booking.

Graphical Maps

For each map, two representations are given, as shown in Fig. 5.6: tabular and graphic format.

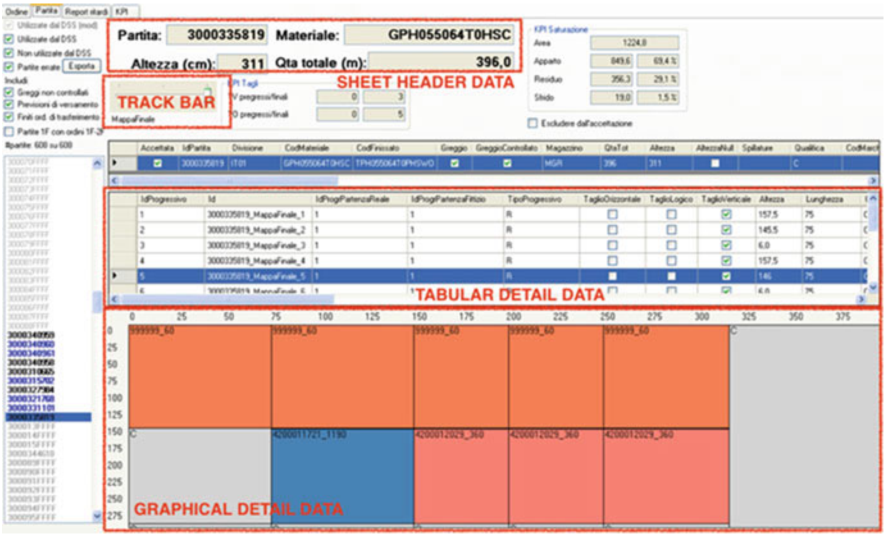


Fig. 5.6 UI to show the sheet details

The tabular format is a reproduction of how the data is represented in the custom SAP program used for the booking process before the DSS. It consists of a table where each row contains the information (size, position, quality level, booked order) of a single stock-piece.

The graphic format under the SAP-style data is a scale representation of the map. Grey rectangles represent stock-pieces not assigned to orders; the label on the top-left corner identifies the quality level of each stock-piece. Booked stock-pieces are colored rectangles whose top-left corner is occupied by order code.

Browsing through the track bar, the user can have at a glance how the evolution of the sheet, from its initial state to its final one proposed by the DSS optimization engine. Furthermore, having both the familiar SAP-style representation in parallel with the graphical one allows the user to obtain both synthetic, fast information on the sheet usage along with the fine-grained detail needed for peculiar cases.

Orders vs Sheets

From the orders point of view, the entire set is displayed by default in a list box, whose elements can be filtered, in order to analyze only a subset of orders. Two types of filter are available: by department associated to each order and by booking state; the latter allows to select orders totally assigned or already assigned before booking process or not assigned for various reasons (lack of material, bad use of fabric sheets, unacceptable trim loss, incorrect order).

When the user selects a single element of the list box, header information of order are shown, together with the list of fabric sheets booked to satisfy the order. Moreover, for each available fabric sheet and for each stock-piece which composes it, the DSS verifies if all conditions for order booking are respected and eventually points out those ones which forbid the assignment.

An important and very common way the user adopts to analyze the solution proposed by the DSS is to select an order, see on which sheets it is booked, double click on one of them to go to the sheet-centered part of the UI to view the graphical map of the sheet, then go back to the order, check another sheet it is booked on and iterate through this pattern over and over. The UI was developed to make this continuous hops between the order and the sheet point of view as easy and fast as possible. With this objective in mind, whenever an order ID is displayed in the UI, if the user double-clicks it, the detail view of that order is shown. On the other way round, wherever a sheet ID is displayed in the UI, if the user double-clicks it, the detail view of that sheet comes up.

KPIs

Next to the indicators referred to orders, fabric sheets and delays which allow only a punctual evaluation of the solution, global KPIs are given to summarize different aspects of booking process such as the number of satisfied orders, total satisfied quantity, the number of booked fabric sheets, total trim loss, total delay, the number of horizontal and vertical cuts.

Supported by various representations of final results, the user is now able to decide to accept entire solution or a part of it, by selecting the set of fabric sheets to be included. Thus scheduling of orders and updated maps of fabric sheets can be reported into output text files, ready to be sent to information system.

5.5 System Implementation

The dominant DSS component is its decision engine, whose function consists of determining how to split the production of orders onto available fabric sheets, under assignment and user-defined constraints.

The decision engine of Booking DSS uses a two-phases technique combining a pattern generation algorithm followed by a multi-objective ILP model. For each fabric sheet, a pattern is defined as a possible allocation of part of an order to the considered stock-piece; in other words, a pattern is a possible way to use a stock-piece. Each pattern is related to a single order and can be horizontally replied.

5.5.1 Problem Statement

The main decision problem the DSS faces is a very domain-specific variant of the well known CSP (Cutting Stock Problem), in which, in general, a set of smaller items, characterized by various sizes and shapes, needs to be produced through a cutting process, starting from a set of larger objects (stocks) whose sizes and availability are also known.

The cutting stock problem was introduced by Gilmore and Gomory [7, 8]. Several variants of CSPs were addressed by the literature in the last decades: a widely adopted typology of such problems can be found in [5, 6]. A well-known variant of the classical CSP concerns the 2DCSP (2-Dimensions CDP), where a set of orders configured by bi-dimensional items should be cut from a set of larger rectangular stocks.

A first attempt to extend the CSP to multiple dimensions was made in [9], where a variant of the CSP was introduced, considering orders of rectangular items to be cut from rectangular stock pieces. The 2DCSP can involve different problem variants depending on several aspects such as the shape of the orders to be cut, as well as the technological constraints to be fulfilled. A classification criterion concerning 2DCSP problems is presented in [20], where problems are assessed according to the shape of the orders to be cut (regular vs. irregular and, among regular shapes, rectangular vs. non-rectangular), as well as in relation to the cutting related constraints (namely, guillotine and orientation constraints).

The decision problem addressed by the DSS is a variant of the 2DRCSP (2-Dimension Rectangular CSP) including both guillotine and orientation cutting constraints. A survey of the main researches regarding the CSP can be found in [1], while more specific literature on the 2DRCSP can be found in [19].

Many recent papers address industrially-relevant variations of the 2DCSP. A typical industrial application of the 2DCSP is presented in [4], where a sequential heuristic procedure based on pattern generation is benchmarked against five published alternative algorithms. An exact algorithm for the N-sheet 2DCSP is presented in [18], where a strong hypothesis is that the available sheets have all the same size. The size variability of the available stocks is heuristically addressed in [10]: the hybrid approach that exploits a combination of several techniques (Best Fit, backtracking, simulated annealing and binary search) outperforms many other existing algorithms.

In these papers, however, as well as in the largest part of the literature concerning CSP, a fundamental assumption is made: that the size and the quantities of the required items are well known. In the decision problem at the core of the presented DSS, on the contrary, much of the possibilities to optimize the production relies on the accepted tolerances on both the required quantities and sizes of the items. A stochastic approach is used in [14] to address uncertainty on the values of items, while [11] and [12] address uncertainty on the items and stocks sizes using fuzzy sets theory, but both approaches do not match the needs of the addressed industrial case.

A detailed description of the mathematical programming model is reported in Appendix.

5.5.2 Pattern Generation

An example of pattern generation referred to a specific fabric sheet is described below. The fabric sheet is 160 cm high and has an initial map made up of two stock-pieces characterized respectively by quality level B1 and B2 and length equal to 100 and 120 m. The set of orders to be produced is composed by three elements, whose required quantity, height and cut length are known and reported as follows:

Order	Quantity (m)	Height (cm)	Cut length (m)
O_1	[180, 200]	[76, 80]	[50, 60]
O_2	[60, 70]	[158, 162]	[30, 100]
O_3	[100, 120]	[152, 156]	[50, 50]

Supposing that all orders can be allocated to the considered fabric sheet without violating any user-defined constraints, for stock-pieces with quality level B1 and B2, Booking DSS generates the patterns represented respectively in Figs. 5.7 and 5.8.

5.5.3 ILP Model

Multiple orders can be allocated to the same stock-piece only combining different patterns defined on the same stock-piece. For this purpose, an optimization model chooses which patterns to be produced and, if they can be replicated more than once in the stock-piece length, how many times they must be produced. Although the main objective is the minimization of total delay, trim loss is also an issue to be considered. In order to pursue the double objective of minimizing trim loss while minimizing total delay, two models are solved: the former to minimize total delay, the latter to minimize trim loss, while allowing an acceptable degradation in the total delay with respect to the minimum achieved by the former model.

After this two-phases procedure, the intermediate map is constructed. A possible solution referred to the described example is reported in Fig. 5.9a. Here, a first pattern allocates two rectangular portions of order O1, vertically piling them on

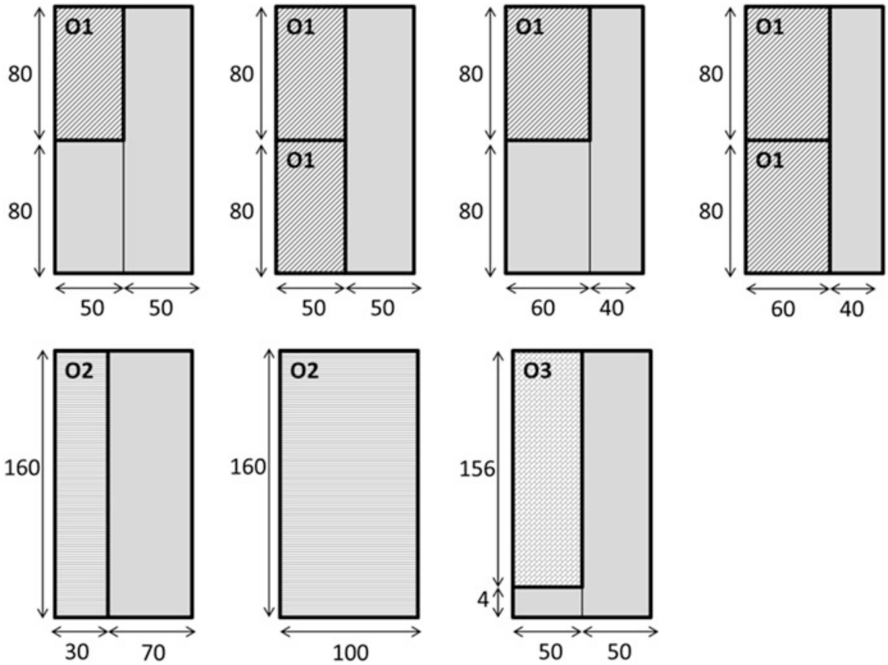


Fig. 5.7 Patterns for stock-piece B1

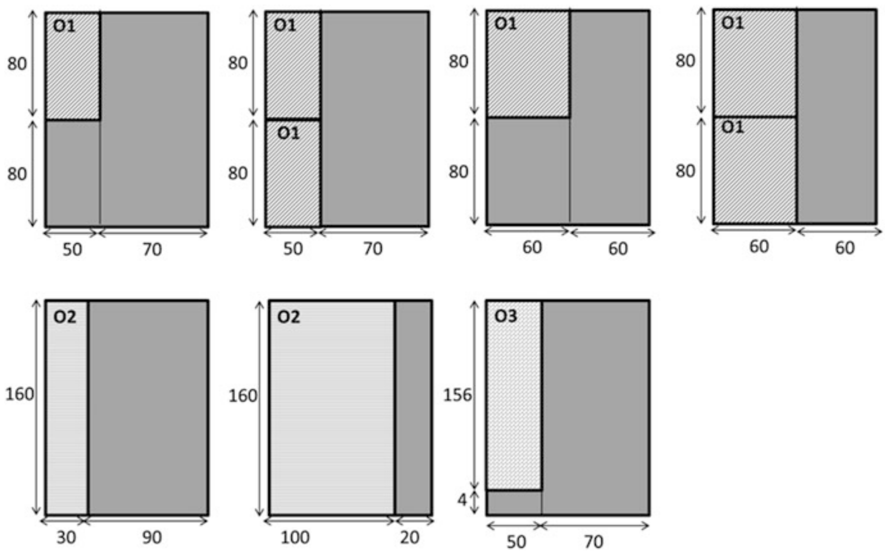


Fig. 5.8 Patterns for stock-piece B2

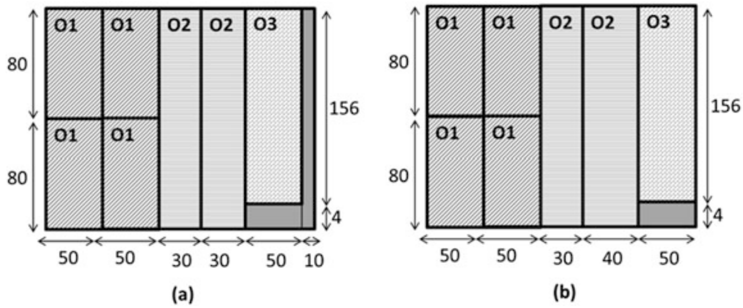


Fig. 5.9 Optimization solutions. (**a**) ILP solution (**b**) post-optimization solution

stock-piece with quality level B1; such pattern is then replicated twice horizontally. Another pattern allocates two portions of order O2 and another one a portion of O3 in stock-piece with quality level B2. The grey areas represent unused parts of the fabric sheet.

In order to produce a further decrease of trim loss, the solution returned by multi-objective model is improved by a post-optimization procedure, which generates final maps. In Fig. 5.9b, the final map of the described example is shown.

5.6 System User Experience

The DSS was designed from the beginning with the intent to be used by the booking employees, so it had to look familiar enough to them, allowing them to be able to adopt procedure that were more or less the ones they followed on the previous system, but, at the same time, the DSS had to incorporate well defined strategies, the ones shared with the supervisor, and should gradually lead the user to follow the implemented strategies rather then trying to force the tool into a specific logic path for each case.

This was achieved by three main aspects of the tool:

- once the set of orders to be processed was defined, the overall booking process is automatic, without requiring the user to waste time individually selecting one fabric sheet at a time;
- reviewing the DSS proposals corresponding to the different scenarios is fast and easy for the user, and the possibility to browse through the used sheets, being able to rapidly evaluate each sheet usage through the graphical representation discussed in Sect. 5.4.2, is a key feature;
- the user is able to obtain specific behaviors through the parameters settings and the user-defined constraints discussed in Sect. 5.4.1, though the evidence of quantitative KPIs on the UI makes it also clear how often what the user thought

was a best option according to a non-formalized strategy rarely translated into the best option in terms of KPIs values.

5.6.1 Lessons Learnt

Having to deal with an established base of many employees with years of experience with a method that was not supported by any computer intelligence, the main challenge was not only to be able to implement the right model to solve the problem at hand, that was a non-trivial task anyway, but also to introduce a new kind of tool to the user without alarming him or making him fear that the machine-based intelligence could replace his function.

Indeed, the user was driven to understand how his unique understanding of the problem was better capitalized letting the DSS taking most of the decisions on the vast majority of the cases that were to be considered as common situations, while his time was better spent managing those less frequent but cases where the DSS' behavior could be better tuned through the help of the user know-how.

5.6.2 System Sustainability

The development of the DSS was a long and challenging task that required an effort that was probably possible only because of the research aspects that justified the participation of academic staff. However, focusing on the software development aspects of the project, its costs were largely paid back considering the following results of the DSS adoption:

- the employees that once were full-time dedicated to the long and time-consuming booking activity now spend just few hours each week supervising the DSS run and their expertise can be more effectively valorized in less repetitive activities;
- the better usage the DSS makes of remainder fabric sheets allowed a drastic reduction of the warehouse size;
- the graphical visualization of the maps allowed a reduction in the number of cutting errors.

A summary of the results achieved during 3 weeks of side-by-side usage of the DSS along with the previous SAP-based procedure is reported in the following table:

Week		SAP-based	DSS	% delta
W1	Total orders quantity	28,437	28,437	
	Booked on available material	10,170	19,534	92.1%
	Available material	278,481	278,481	
	Booked on forecasts	18,247	9086	−50.2%
	Forecasts quantity	224,307	224,307	
	Trim on available material	1988	1622	−18.4%
	QDL	1,946,219	1,661,080	−14.7%
W2	Total orders quantity	32,198	32,198	
	Booked on available material	2929	6212	112.1%
	Available material	56,676	56,676	
	Booked on forecasts	28,462	26,431	−7.1%
	Forecasts quantity	148,517	148,517	
	Trim on available material	482	359	−25.5%
	QDL	514,362	144,984	−71, 8%
W3	Total orders quantity	29,351	29,351	
	Booked on available material	1546	1592	3.0%
	Available material	33,431	33,431	
	Booked on forecasts	30,219	28,288	−6.4 %
	Forecasts quantity	150,351	150,351	
	Trim on available material	265	220	−17.0 %
	QDL	1,756,905	1,043,576	−40.6%

For each week, the following quantities are reported:

- the total amount of square meters required by the orders to be produced (the same for both approaches)
- the amount of the required quantity that was booked on available material (rather than onto forecasted arrivals)
- the total amount of already available material (the same for both approaches)
- the amount of the required quantity that was booked on forecasted arrivals
- the total amount of forecasted arrivals (the same for both approaches)
- the trim produced by bookings made on available material
- the QDL (Quantity by Day Lateness) as detailed in Appendix, is a quantitative measure of how good the solution is in terms of the delay minimization objective

The last two items are the actual KPIs used by the decision model, while the preceding ones had sense for the supervisor to better evaluate the quality of the results: booking more on available material, indeed, is mainly due to a better usage of the remainders of previous cutting processes that in the SAP-based procedure where less likely to be used.

5.6.3 System Upgrade and Maintenance Issues

The DSS was initially designed with the only task to address the booking process as it was done by the booking employees, with the trim loss as the only driver for the decisions.

Once the DSS was adopted and its usage completely replaced the former procedure, its KPI-based reasoning led the firm management to express the need for an important evolution of the DSS to consider the booking process from a wider and more strategic point of view. The booking process was now required to consider, on the same KPI-centered base, both the traditional production-related aspects and the *new* customer-oriented issues.

The second major version of the DSS is currently used as the main decision tool in the production planning activities for the cutting division of the firm. Its success has also lead to a further development: a new DSS was required by the company, this time to address the looms production planning. The deep knowledge of the firm's production cycle matured with the development of the two versions of the DSS is helping defining the main strategic aspects of the looms production planning problem, though it is very different from the previous one.

The evolution of the DSS over time was a proof that the adopted approach, based on a deeply customizable pattern generation algorithm followed by a general ILP mathematical model, was the right trade-off between the need to model very company-specific needs and the ability to use a general mathematical and quantitative approach to evaluate different solutions. Moving to the second major version of the original DSS, though providing the company with a much more powerful and strategic decision tool, almost only required changes in the mathematical model. On the other hand, the looms production planning tool is requiring a deeper effort to adapt the pattern generation module, while the mathematical model should not be significantly varied.

Another important and very reusable part of the DSS was the graphical map viewing tool. The effectiveness of the graphical representation of the fabric sheets used by the booking DSS, indeed, lead to a further development. The final map of each fabric sheet, including all the reservations made on it, can be of great help during the actual cutting phase, helping the cutting employee rapidly understand how the sheet needs to be cut. An offspring of the DSS is a visualization software that now runs on the PC close to each cutting machine, that constantly shows the graphical map of the sheet that is being processed.

Finally, the choice to keep the DSS well separated from the information system was helpful not only during the early stages of the DSS adoption, allowing its side-by-side usage with the previous procedure, but it also demonstrated its effectiveness during the many years of maintenance of the DSS. Each time a problem arises with the usage of the DSS, the maintenance procedure takes place starting from the local database the DSS uses: a copy of it is backed up when the problem happens, and all the debug activity is remotely conducted using this snapshot of all the data that were considered by the DSS during its run.

5.7 Conclusions

We presented a DSS test case from a high-tech fabrics company located near Como, Italy. A long-standing collaboration between our academic institution and the company has lead not only to the development of a production planning DSS, but also to its evolution to include larger and more strategic aspects and to the development of other decision tools to support other parts of the company's production.

A long and thorough iterative design and development phase in the early stages of the DSS project was one of the key aspects that helped the success of the DSS, along with the choice to always involve its target users during the requirements gathering and afterwards and with the decision to make the DSS available as a stand-alone tool, that could be initially used on subsets of data and then gradually fed with larger amounts of data. The effectiveness of the mathematical model underlying the DSS and the usage of a sophisticated but extremely friendly UI were also key elements of the project.

Acknowledgements We thank the SAATI SpA staff that was involved in the DSS project, whose help and open-minded approach was fundamental for the success and evolution of the project. We especially thank the Supply Chain Manager, Raimundo Mediavilla, and the IT Manager, Franco Lonati, for their support. This work was supported by Progetto PON/MIUR Vis4Factory: Sistemi Informativi Visuali per i processi di fabbrica nel settore dei trasporti—Codice Progetto: PON02_00634_3551288

Appendix: Mathematical Model

Most works on the 2D Rectangular CSP issued by the scientific literature deal with the case in which the orders to be cut are rectangles of well-known sizes, i.e. each item to be produced is a rectangle with given height and width. This is a perfectly reasonable assumption for real industrial applications and, in particular, when the items obtained by the cutting process represent the final products flowing through the supply chain. However, when the production process involves earlier stages of the supply chains, i.e. those concerning the production of semi-finished products, each order (in terms of rectangular items) issued to a given supplier is typically characterized by two parameters: the quantity of items and the size of each item, both expressed in terms of ranges within a given tolerance.

As an example, an order for a fabric supplier could require an overall quantity of $4900 \div 5100$ m, while the size of each rectangle is accepted if its width is within $120 \div 130$ cm and its length is within $48 \div 52$ m.

Several technological constraints must be considered:

- the manufacturing process requires a stock of raw material to be finished (in terms of color and other thermo-physical treatments) before it can be cut; a single stock of raw material can only serve orders that require the same finish treatments;

- for logistical and production reasons, stocks are not immediately available to be cut, but they can be used after a certain lead time, expressed in days necessary to collect them from the place where they are located and, in case of ready to be finished or free raw stocks, to complete all needed treatments; an availability date is associated with each stock, with reference to the date when it will be ready to be cut;
- each stock of raw material is checked for quality and can then be split into stock pieces characterized by a homogeneous quality; orders can require a minimum quality.
- orders require the overall quantity (within the given tolerance) by a final due date. In most cases, due to the large space required by the final product, the producer contracts with the customer for partial intermediate shipments; so, for each order, a set of promised shipments is defined, in terms of quantities to be shipped at given due dates.

The decision maker has to manage two main aspects: how to split the production of the customers orders onto available stocks and how to finish the free raw stocks, with multiple objectives and constraints, namely: respect the given quantity and size range constraints, respect real world constraints, minimize the trim loss and *optimize* the due dates requirements.

The problem to be solved is a variant of the 2DRCSP with guillotine and orientation constraints, in which the items to be produced are correlated by production orders that can include rectangular items having different sizes, such sizes can be part of the decision process, as long as they respect the given tolerances and multiple objectives related to trim loss and due dates requirements need to be pursued.

QDL

As reported in the main sections, the booking problem requires to pursue a double objective: both a trim loss and an overall delay minimization.

It is straightforward to define a trim loss and how to *measure* a solution in its terms, but the multiple promised shipments for each order lead to define a Key Performance Indicator that allows to measure a solution in terms of how much it satisfies (or does not satisfy) such shipment requirements.

We define the Quantity by Day Lateness (QDL) as the unsatisfied part of the quantity to be shipped, multiplied by the number of days of lateness. For instance, let us consider an order with the following shipment requirements:

Due date	Required quantity
20/11/2012	100
01/12/2012	400

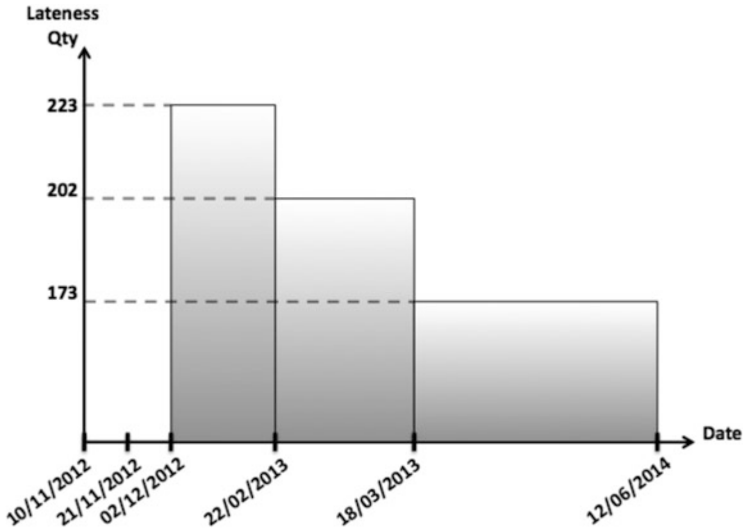
and suppose we plan the following shipments:

Stock	Availability date	Satisfied quantity
S1	21/02/2013	21
S2	09/11/2012	27
S2	09/11/2012	27
S3	09/11/2012	25
S5	09/11/2012	100
S6	09/11/2012	25
S7	17/03/2013	29

How quantities are satisfied through time can be hence summarized by the following table:

Availability date	Satisfied quantity
09/11/2012	277
21/02/2013	21
17/03/2013	29

while a graphical representation of how QDL is calculated is the following:



If all the promised shipment quantities by date are met, QDL will be 0. If an order is entirely fulfilled but with some delay through time, QDL will measure such delay in a rigorous manner QDL also provides a measure of how much of an order is not fulfilled (see date 12/06/2014 in the example). If part of an order is not fulfilled, an estimation of how late at most it will be fulfilled is used to simulate its fulfillment and hence evaluate the QDL.

Minimizing the QDL, hence, *optimizes* the due dates fulfillment in the sense that it: maximizes the amount of orders where all shipment promises are met, minimizes the overall delay of orders that can be entirely fulfilled but not exactly matching the promised shipments and minimizes the amount of unfulfilled orders.

ILP Model

An ILP model is introduced in order to choose which patterns to be used for production and how much to produce of each pattern. Such decision implicitly determines which finish will be made on each used stock. The main objective is the minimization of the total QDL, while a secondary goal is the minimization of the trim loss.

The following mathematical formulation is used:

$$\min \sum_{k\in K}\sum_{d_1\in D_k}\sum_{d_2=d_1+1} n_{d_1 d_2} w_{k d_2}$$

s.t.

$$\sum_{i\in I_p} x_i \le L_p \qquad \forall p \in P$$

$$\sum_{i\in I_k} m_{ik} x_i \le l_k + lt_k^+ \qquad \forall k \in K$$

$$x_i \le l_i z_i \qquad \forall i \in I$$

$$x_i \ge z_i \qquad \forall i \in I$$

$$\sum_{i\in I_{sf}} z_i \ge t_{sf} \qquad \forall s \in S_r, \forall f \in F$$

$$\sum_{i\in I_{sf}} z_i \le |I_{sf}| \cdot t_{sf} \qquad \forall s \in S_r, \forall f \in F$$

$$\sum_{f\in F} t_{sf} \le 1 \qquad \forall s \in S_r$$

$$x_i \le l_i^{max} v_i \qquad \forall i \in I$$

$$x_i \ge l_i^{min} v_i \qquad \forall i \in I$$

$$\sum_{\substack{d'\in D_k \\ d'<d}} \sum_{i\in I_{kd'}} m_{ik} x_i + w_{kd} \ge \sum_{\substack{d'\in D_k \\ d'<d}} l_{kd'} \qquad \forall k \in K, \forall d \in D_k$$

$$x_i \ge 0 \qquad \forall i \in I$$

$$z_i \in \{0,1\} \qquad \forall i \in I$$

$$z_i \ge 0\ integer \qquad \forall i \in I$$

$$t_{sf} = \{0,1\} \qquad \forall s \in S_r, \forall f \in F$$

$$w_{kd} \ge 0 \qquad \forall k \in K, \forall d \in D_k$$

QDL Minimization Objective

The index k denotes an order. D_k is the sorted set of dates involved in the computation of QDL for order k, while $n_{d_1 d_2}$ is the number of days between consecutive dates d_1 and d_2. The non-negative decision variable w_{kd} is the difference between required and satisfied cumulative quantities for order k until date d if this difference is a positive value, zero otherwise.

Stock Piece Maximum Length Usage

The index p denotes a stock piece. $i \in I_p$ is the set of patterns generated for p. x_i is a non-negative variable that represents the amount (length) of pattern i that the model decides to produce. The length occupied by all the patterns produced on p cannot exceed its length.

Order Required Quantity

I_k the set of patterns referred to order k. m_{ik} is the layer multiplicity of order k in pattern i. The overall length served for each order k cannot exceed the total required quantity l_k increased by the positive allowed tolerance.

Cut-Lengths

Integer variable v_i is introduced to force variable x_i to assume values that are multiple of at least one of the cut-lengths in the $[l_i^{min}, l_i^{max}]$ range.

Finish

I_{sf} is the set of patterns generated from fabric reel s with finish f, with S_r set of fabric reels and F set of all possible finishes. If a certain finish is assigned to a fabric reel, it must contain at least a pattern generated from an order which requires that finish. On the other hand, if a finish is not associated to a fabric reel, it cannot contain any pattern generated from orders which require that finish. Finally, a single finish is allowed per reel.

Secondary Objective

The following variant of the first model is used to pursue the secondary objective to minimize the trim loss, losing not more than Λ over the optimal QDL^* value.

Binary variable y_s is equal to one only if reel s is used.

$$
\begin{aligned}
&\min \sum_{s\in S}\sum_{i\in I_s}(A_s - A_i x_i) - \sum_{s\in S} A_s(1-y_s) \\
&\text{s. t.} \\
&\sum_{k\in K}\sum_{d_1\in D_k}\sum_{d_2=d_1+1} n_{d_1 d_2} w_{k d_2} \le QDL^* + \Lambda \\
&\sum_{i\in I_s} z_i \ge y_s \qquad\qquad \forall s \in S \\
&\sum_{i\in I_s} z_i \le |I_s| \cdot y_s \qquad\qquad \forall s \in S
\end{aligned}
$$

References

1. Cheng, C.: The cutting stock problem — a survey. Int. J. Prod. Econ. **36**(3), 291–305 (1994)
2. Chung, L., Do Prado Leite, J.: On non-functional requirements in software engineering. In: Conceptual Modeling: Foundations and Application, pp. 363–379. Springer, Berlin (2009)
3. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: Non-Functional Requirements in Software Engineering. International Series in Software Engineering. Springer, Berlin (2012)
4. Cui, Y.P., Cui, Y., Tang, T.: Sequential heuristic for the two-dimensional bin-packing problem. Eur. J. Oper. Res. **240**(1), 43–53 (2015)
5. Dyckhoff, H.: A typology of cutting and packing problems. Eur. J. Oper. Res. **44**(2), 145–159 (1990)
6. Dyckhoff, H., Finke, U.: Cutting and Packing in Production and Distribution: A Typology and Bibliography. Springer, New York (1992)
7. Gilmore, P.C., Gomory, R.E.: A linear programming approach to the cutting-stock problem. Oper. Res. **9**(6), 849–859 (1961)
8. Gilmore, P.C., Gomory, R.E.: A linear programming approach to the cutting stock problem-Part II. Oper. Res. **422**, 863–888 (1963)
9. Gilmore, P.C., Gomory, R.E.: Multistage cutting stock problems of two and more dimensions. Oper. Res. **13**(1), 94–120 (1965)
10. Hong, S., Zhang, D., Lau, H.C., Zeng, X., Si, Y.W.: A hybrid heuristic algorithm for the 2D variable-sized bin packing problem. Eur. J. Oper. Res. **238**(1), 95–103 (2014)
11. Kim, J.-K., Lee-Kwang, H., Yoo, S.W.: Fuzzy bin packing problem. Fuzzy Sets Syst. **120**(3), 429–434 (2001)
12. Nasibov, E.N.: An algorithm for constructing an admissible solution to the bin packing problem with fuzzy constraints. J. Comput. Syst. Sci. Int. C/C Tekhnicheskaia Kibernetika **43**(2), 205–212 (2004)
13. Parker, C.: An approach to requirements analysis for decision support systems. Int. J. Hum. Comput. Stud. **55**(4), 423–433 (2001)
14. Perboli, G., Tadei, R., Baldi, M.M.: The stochastic generalized bin packing problem. Discret. Appl. Math. **160**(7), 1291–1297 (2012)
15. Power, D.J.: Supporting decision-makers: an expanded framework. Proc. Inf. Sci. IT Educ. **1**, 1901–1915 (2001)

16. Power, D.J.: Decision Support Systems: Concepts and Resources for Managers. Quorum Books, Westport (2002)
17. Power, D.J., Sharda, R.: Model-driven decision support systems: concepts and research directions. Decis. Support Syst. **43**(3), 1044–1061 (2007)
18. Steyn, T.: An exact algorithm for the N -sheet two dimensional single stock-size cutting stock problem. Orion **31**(2), 77–94 (2015)
19. Suliman, S.: A sequential heuristic procedure for the two-dimensional cutting-stock problem. Int. J. Prod. Econ. **99**(1–2), 177–185 (2006)
20. Wäscher, G., Hausner, H., Schumann, H.: An improved typology of cutting and packing problems. Eur. J. Oper. Res. **183**(3), 1109–1130 (2007)

Chapter 6
A Decision Analytical Perspective on Public Procurement Processes

Mats Danielson, Love Ekenberg, Mattias Göthe, and Aron Larsson

Abstract If procurement processes are to be taken seriously, purchase managers need decision support tools beyond those that only ascertain that the formal requirements are met. This chapter demonstrates some fundamental flaws with common models used in procurement situations, flaws that are so serious that the evaluations of tenders often become meaningless and may lead to large and costly miscalculations. We demonstrate how the equitability of the tender evaluations can be significantly improved through the use of multi-criteria decision analysis with numerically imprecise input information. Due to this, the computational part of the evaluation step becomes more complex, and algorithms targeted for decision evaluation with imprecise data are used. We therefore present a procurement decision tool, *Decide*IT, implementing such algorithms that can be used as an instrument for a more meaningful procurement process. Of importance is to allow for a more realistic degree of precision in the valuation and ranking of tenders under each evaluation criterion, as well as the associated weighting of the criteria, since the criteria are often of a more qualitative nature. Through this, both quantitative and qualitative statements could be easily managed within the same framework

M. Danielson
Department of Computer and Systems Sciences, Stockholm University, Kista, Sweden

International Institute for Applied Systems Analysis, Laxenburg, Austria
e-mail: mats.danielson@su.se

L. Ekenberg (✉)
International Institute for Applied Systems Analysis, Laxenburg, Austria

Department of Computer and Systems Sciences, Stockholm University, Kista, Sweden
e-mail: ekenberg@iiasa.ac.at

M. Göthe
Department of Computer and Systems Sciences, Stockholm University, Kista, Sweden
e-mail: mattias@tanto-system.se

A. Larsson
Department of Computer and Systems Sciences, Stockholm University, Kista, Sweden

Department of Information and Communications Systems, Mid Sweden University, Sundsvall, Sweden
e-mail: aron@dsv.su.se

© Springer International Publishing Switzerland 2016
J. Papathanasiou et al. (eds.), *Real-World Decision Support Systems*,
Integrated Series in Information Systems 37,
DOI 10.1007/978-3-319-43916-7_6

and without the need to introduce ad-hoc and often arbitrary conversion formulas supposed to capture the trade-off between criteria.

6.1 Introduction

The values at stake in public sector procurement are very high, and therefore public procurement is an important issue for policy makers, suppliers and the general public. For instance, public procurement in Sweden alone has an annual turnover of between 16 and 18 % of the gross domestic product and similar figures are found in other of the countries within the OECD.[1] In the European Union (EU), this is in part a consequence of a pro forma comprehensive EU regulatory framework, with the double ambition of both increasing the efficiency and dynamics of the free European market, increasing transparency and predictability, and to avoid corruption. However, procurement processes as they are conducted in practice are often far from satisfactory from the perspectives of the buyers, the suppliers, and that of the general public.

As a consequence, in Sweden the confidence in the effectiveness of public procurement is relatively low within the suppliers,[2] and there have been some calls for dismantling of the entire legal framework for procurement in political circles.[3] It is commonplace that suppliers challenge the processes in costly and lengthy appeals, but also that the buyers are forced to start the entire process over when the effects of an accepted tender become apparent. These problems have of course a bearing on unnecessary financial costs, but also on the efficiency of policy implementation at large, and the public's confidence in the government in general.

The underlying rationale behind the current chapter is that we believe that the reasons for the shortcomings can, at least in part, be attributed to the methods employed to evaluate the bids when awarding a contract to a winning supplier since from a decision analytic perspective the procurement methods are unsatisfactory at best. The problems emanate to a large extent from three key issues, namely that those responsible for procurement processes often

(1) require unrealistic precision
(2) deal with qualitative values in an erroneous way, and
(3) manage value scales incorrectly.

Procurement in general is a large and growing field of research, but the vast majority of studies has been on private sector procurement, see e.g. [15]. Since the 1960s, the structured use of multiple criteria assessments when selecting a supplier has become more and more common, with a growing number of evaluation methods

[1] http://www.oecd.org/gov/ethics/public-procurement.htm.

[2] http://handelskammaren.net/sv/Nyheter/Nyhetsarkiv/Pressmeddelanden/2015/mars/lagt-fortroende-for-offentlig-upphandling-bland-vastsvenska-foretag/ (in Swedish).

[3] http://www.dagenssamhalle.se/debatt/skrota-lou-i-sjukvarden-134 (in Swedish).

proposed. An extensive review of methods can be found in [2]. In the public sector as compared to procurement in the private sector, there are special requirements and circumstances that make the process even more complex. One key difference is that in the public sector, both stakeholders and objectives are more diverse and possibly conflicting. Another difference is that public procurement is not merely a way to acquire goods or services, but also an important tool for policy makers, for example as an instrument for driving innovation from the demand-side or promoting environmental or social values [21].

However, the main formal difference between private and public sector procurement is that a buying entity in the public sector is typically regulated by a more extensive legal framework than a corresponding entity in the private sector. This paper deals to some extent with the limitations of the current European legislation on public sector procurement from a decision analysis perspective, and that subject has been the focus of studies like [15, 25], and also in [1], where a majority of 189 public procurement processes was found to be using deficient award mechanisms.

A procuring entity within the public sector "must provide the best value for money in public procurement while respecting the principles of transparency and competition".[4] Award decisions can be made using the lowest price criterion or a combination of qualitative and quantitative aspects (most economically advantageous tender—MEAT).

In the literature, the evaluation of suppliers in procurement is commonly referred to as "tender evaluation" or "supplier selection". A review of tender evaluation and selection methods suggested in the OR literature is provided in Ref. [2], however not restricting the scope to public procurement and its regulations and limitations. Most of the proposed methods published in the OR literature instead focus on supplier selection in for-profit manufacturing companies supply chain management, such as the reviews in [2, 16, 26]. When delimiting the context to EU public procurement, less studies have been made. Typically in such contexts, the buyer can base the selection based upon highest quality (aggregating the quality aspects), lowest price (of those qualifying with respect to quality), or an aggregation of price and quality searching for the economically most advantageous tender (MEAT).

There is a plethora of evaluation models available if several criteria are to be evaluated simultaneously. Some rely on a formal quantification of the quality dimension of the bid, such as using well-defined proxy measures, for example number of employees with a university degree or number of previous relevant contracts won. Other models define a minimum level of quality, and select a winner from the bidders passing the minimum level using price as the single criterion. Depending upon the procurement context and the information available, i.e. to what extent the cost of quality is known, different tender evaluation approaches are proposed. If the cost of quality is well known and many competitors can offer optimal quality, it is claimed that lowest price is the proper approach and not MEAT [1].

[4] http://ec.europa.eu/growth/single-market/public-procurement/rules-implementation.

In MEAT situations, there exists a large variety of scoring formulas aimed to aggregate price and quality aspects. Some methods evaluate each offer within a criterion using a point scale, weighing the points using the weights of each criterion to reach a score. The different models can give different outcomes when applied to the same bid, but can still meet the authorities requirements for transparency and predictability.[5] In [23] 38 different scoring formulas and their usage for MEAT tender evaluations is analysed, indicating that the priority weights put on quality and price is dependent on the selection of scoring formula and that many widespread scoring formulas are overly sensitive to price. A fundamental property of the scoring formula used is however buyer preference consistency, meaning that the top-ranked supplier as advocated by the scoring formula also shall meet the preferences of the buyer better than the other suppliers. In order to ensure consistency, Ref. [20] advocates that instead of setting a score on price, setting a price on quality. The problem with this approach is that it is cognitively demanding, and in more complex procurement situations it becomes an utterly delicate activity that the buyer must be capable of.

From the viewpoint of decision science, the award stage in a procurement process is a multi-criteria decision problem. There are several approaches to multi-criteria decision making, the key characteristic being that there are more than one perspective (criterion, aspect) to view the alternatives and their consequences from. We have during the latter decades developed various computational methods for formal decision analysis with imprecise information also implemented in toolkits for multi-criteria decision making (see, e.g., [4, 7, 8, 10]) and will below discuss how to apply some aspects of these on the problems addressed above. The system described in this chapter is essentially a model driven DSS. It supports procurement decisions by making a model of the alternative actions available in an actual procurement decision situation and modelling the possible selection of each supplier as one distinguishable course of action in a decision tree.

6.2 Decision Analysis for Procurement

As previously mentioned in the former section, there may be several reasons why a certain procurement process fails and ends up in an unsatisfactory evaluation result, but in many cases, failure can be attributed to the methods employed to evaluate the bids when selecting a winning supplier. The point-of-departure of this chapter is that the difficulties emanate to a large extent from three key issues in the procurement process.

The first issue has to do with *unrealistic precision* when stating the weights of different criteria in tender documents. The second issue is about assigning and comparing *qualitative values* in a somewhat naïve way, without ample regard to the

[5]Att utvärdera anbud - utvärderingsmodeller i teori och praktik, Rapport 2009–10, Konkurrensverket (in Swedish).

profound difficulties of, say, comparing monetary, ethical and aesthetic values. The third and final issue is the problem of managing *value scales* without the relevant technical understanding. Below, we explain each of these in some detail.

Assume that we intend to negotiate a contract for a consultancy service in the form of an interior designer. The procurement process is typically divided in three stages. The first stage is creating *specifications* about what the service should consist in, what specific tasks that should be performed, what evaluation criteria we will use and the relative weights of these. The second stage is the *selection* of a set of suppliers meeting the specifications. The third and final stage is deciding which submitted tender is the most preferred one, and *awarding* a contract to the winning tender. When awarding the contract, we will look at several criteria as defined in the specification.

Monetary cost is one of these, but not the only one. Using multiple criteria in this way is common practice in public procurement, and is for example in accordance with the international GPA (Agreement on Government Procurement) treaty, where the European Union is a party. Employed criteria could, apart from price/cost, be, e.g., quality, technical merit, aesthetic and functional characteristics, environmental characteristics, operational costs, cost effectiveness, service costs, technical assistance, delivery date and delivery periods, period of completion, social considerations, sustainability and level of innovation.

We will in this example be using four different criteria, price being one of them. The criteria that have been established are cost, competence, responsiveness, and design concepts, further specified like this:

- Cost—the full monetary cost of the service, divided into hourly rates.
- Competence—mainly how well experienced and/or educated in the field the contractor is.
- Responsiveness—we will conduct interviews with the potential contractors, and make an assessment of the suppliers responsiveness to the relevant demands.
- Design concept—each supplier is supposed to describe how the task could be carried out, and the description will be evaluated in terms of creativity, style, level of innovation, etc.

We have four bids from suppliers *A*, *B*, *C* and *D*, all of which have submitted well-prepared tenders. When using several criteria in the award process, we need some way of expressing the importance of a certain criterion compared to another. The GPA treaty and several corresponding national regulations state that this "relative importance" of each criteria also should be presented to the potential suppliers, for transparency reasons.[6] According to EU regulations, if the authorities use different criteria when evaluating tenders, "each applicant should be informed of the different weighting given to the different criteria (for example price, technical characteristics and environmental aspects)".[7] There are several ways of stating

[6]https://www.wto.org/english/docs_e/legal_e/rev-gpr-94_01_e.htm.

[7]http://europa.eu/youreurope/business/public-tenders/rules-procedures/index_en.htm.

the relative importance. The perhaps most straightforward method is to give each criterion a numerical weight. Further, weights may be assigned a range instead of a fixed number, where the application of such criteria weight intervals is up to the contracting authority.

A common method is to first split the weights between price and quality, and then to further specify quality using sub-criteria. When it is not possible to provide weights that are based on objective measures or proxy values, the criteria can be listed in descending order of importance, a mere ordinal ranking. In our case, we assign each criterion a percentage. We assume the following weights:

- Cost is 40 %
- Competence is 30 %
- Responsiveness is 20 %
- Design concept is 10 %

In our example, we will classify the bids using a five level scale within each criterion. In the tender documents, the scale is described as follows:

5 Much better than the criterion base level
4 Better than the criterion base level
3 Meets criterion base level
2 Somewhat worse than the criterion base level
1 Not corresponding to the criterion base level

When the tenders from the four suppliers are evaluated using this scale, we get the matrix in Table 6.1.

A direct observation is that no supplier strictly dominates another in the sense that it has a higher value than any other for some criterion, and is equally good for all the remaining ones. We therefore cannot directly select a winner without a continued analysis. The next step is to multiply the values for each criterion with the corresponding weight, and add the result to get a weighted sum total. In our case, this becomes:

$$V(A) = 0.4 \cdot 5 + 0.3 \cdot 2 + 0.2 \cdot 2 + 0.1 \cdot 4 = 3.40$$

$$V(B) = 0.4 \cdot 4 + 0.3 \cdot 4 + 0.2 \cdot 3 + 0.1 \cdot 3 = 3.70$$

$$V(C) = 0.4 \cdot 2 + 0.3 \cdot 3 + 0.2 \cdot 5 + 0.1 \cdot 1 = 2.80$$

$$V(D) = 0.4 \cdot 1 + 0.3 \cdot 5 + 0.2 \cdot 2 + 0.1 \cdot 5 = 2.80$$

Table 6.1 Evaluation of tenders

Supplier	Cost	Competence	Responsiveness	Design concept
A	5	2	2	4
B	4	4	3	3
C	2	3	5	1
D	1	5	2	5

Using this way of evaluating, we should award the contract to supplier *B*. Now, the obvious question is whether we made the right choice or not. The answer is simply that we cannot know for sure. The model has too many short-comings to be of any substantial guidance in this respect. In the introduction of the paper, we indicated that there are three fundamental problems pertaining to the model as it is defined above.

6.2.1 Unreasonable Precision

It is not always (indeed, rarely) possible to specify weights with any higher degree of precision. When making non-formalized, everyday choices, we routinely employ weighing of different criteria, but almost never with a precision close to a fixed percentage. When, for example, buying a new car, we typically use several criteria for selecting a make and model, such as price, comfortability, design, social value, etc. We typically rank one criterion equal to, lower than or higher than another criterion, perhaps with a qualifying "much higher/lower" than, but that is about the level of precision we can expect in an everyday situation. In the present context, where stakes could be substantially higher than in the everyday scenario, we need a more formalized and transparent decision procedure relying less on intuition. However, to avoid unrealistic precision, the procedure should preserve some level of imprecision. Using fixed percentages is not a good representation of how we normally deal with several criteria simultaneously, and therefore the model should allow for some kind of fuzziness, such as ranges.

The EU legislation[8] also recognizes the difficulty of assigning precise numerical weights and allows for ordinal rankings of the criteria. However, in real-life procurements, weights are commonly treated as precise statements, even though this is not a legal requirement. This is, in our view, the first of the three systematic mistakes presented above. One way of presenting a mere ranking of the criteria involved is to assign ranges instead of precise percentages. The relative importance of the criterion "Competence" could be presented as "25–35 %", etc. In an "improved" version of the model, we will therefore use ranges instead of fixed percentages. In terms of calculations, it becomes a bit harder to evaluate options with imprecise statements, but there are methods for this as we will see below.

Now assume that the weights of each criterion are the following percentage ranges:

- Cost is 35–45 %
- Competence is 25–35 %
- Responsiveness is 15–25 %
- Design concept is 5–15 %

[8] Article 90 of the EU directive on public procurement, available at http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32014L0024&from=EN.

Table 6.2 Evaluation of tenders using intervals

	Cost	Competence	Responsiveness	Design concept
A	4–5	1–3	1–3	3–4
B	3–4	3–4	3–4	3–4
C	1–3	3–4	4–5	0–2
D	0–2	4–5	1–3	4–5

The difficulties encountered with precise values apply even more to the evaluation of the alternatives within each criterion. In the same way as with the criteria weights, we can preserve a certain level of imprecision and use less precise values by assigning intervals to the alternatives within each criterion (Table 6.2).

As before, we multiply the assigned values with the percentages, but in the improved model we use the sum of the lower-end values multiplied with the lower-end percentages as the lower-end of the sum total range, and vice versa for the higher-end values and percentages. The minimum and maximum aggregated values for A are

$$\min(V(A)) = 0.35 \cdot 4 + 0.35 \cdot 1 + 0.25 \cdot 1 + 0.05 \cdot 3 = 2.15$$

$$\max(V(A)) = 0.45 \cdot 5 + 0.25 \cdot 3 + 0.15 \cdot 3 + 0.15 \cdot 4 = 4.05$$

So for all alternatives we get the following value ranges

$V(A)$ between 2.15 and 4.05

$V(B)$ between 3.00 and 4.00

$V(C)$ between 1.80 and 3.80

$V(D)$ between 1.45 and 3.65

We now see that the situation, not surprisingly, is not as clear-cut anymore. In fact, the lack of precision in the outcome of the improved model only reflects the corresponding property of uncertainty in everyday, intuition-based evaluation using several criteria. It demonstrates that allowing precise values in most cases is an over-simplification that generates unwarranted precision in the result. Intervals are a bit more difficult to assess, in particular regarding the assessments of the qualitative criteria, but at least we can represent and preserve some aspects of the inevitable imprecision in the background information. This is a commendable feature of the model compared to the original version, not a drawback.

6.2.2 Handling Value Scales Over Qualitative Estimates

Using point scales for handling qualitative values can be problematic. If we for some reason cannot use previously agreed-upon measurements or proxies, the evaluation of alternatives must rely on some kind of intuition or subjective sentiment. When subjectively comparing two alternatives with regard to the same qualitative criteria, we can usually without great difficulty state that the one is better than the other. We have a good understanding of what it means to prefer one alternative to the other. But if we also are required to state that preference or "betterness" using, for example, the five-point scale used in the example above, we run into problems. What information, exactly, is a "5" supposed to carry, except that it is better or more preferable than a "4"?

Even if we have some objective measure available, it is of great importance how this measure translates to a point scale. It is important to realize that the ranking order of the alternatives is not the only thing that matters when evaluating alternatives according to a point scale. The fact that our evaluation uses a cardinal scale means that the numerical values used are significant. The actual point values assigned to the different alternatives can be a factor that decides what alternative is ultimately the winner, even if the order between them is kept invariant. Suppose that the suppliers in the example above have presented tenders that rank as follows according to the criterion Cost ("5" being the best, "1" the worst) in Table 6.3.

It would now be reasonable to ask whether we could have constructed the scale a bit differently. For example, it might be more reasonable to give 1 point instead of 2 to the alternative evaluated as "Somewhat worse than the base level", and to adjust the rest of the scale accordingly. With another numerical scale, we might end up with the points in Table 6.4 instead, maintaining the order of the alternatives.

Note that the alternatives are ranked in the same order as before. The only difference is the point value assigned to alternatives *B*, *C* and *D*. If the evaluations

Table 6.3 Ranking of suppliers according to Cost

	Cost	Value
A	Much better than the base level	5
B	Better than the base level	4
C	Somewhat worse than the base level	2
D	Much worse than the base level	1

Table 6.4 Revised ranking of suppliers according to Cost

	Cost	Value
A	Much better than the base level	5
B	Better than the base level	2
C	Somewhat worse than the base level	1
D	Much worse than the base level	0

in the other criteria stay the same as in Table 6.1, we get the following revised point totals:

$$V(A) = 3.40$$
$$V(B) = 2.90$$
$$V(C) = 2.40$$
$$V(D) = 2.40$$

Now supplier A becomes the preferred one. Not paying enough attention to how the cardinality of a point scale can impact the end result will lead to arbitrariness and transparency problems. This is an obvious problem when we are handling more qualitative aspects of the criteria, but in general, the problem is the same for any criteria where there is no linear relationship between the points assigned and an objectively measurable value.

6.2.3 *Deficiencies in the Handling of Value Scales*

In a situation where different criteria are to be valued and weighed against each other using a common measurement such as a ten-point scale, it is important to be aware of the fundamental difficulties of measuring completely different things using the same scale. Even if legal frameworks require weighing of criteria according to a certain method, the question is whether the framework is comprehensive enough to take account for the how applying the value scale may impact the outcome, and in some cases even completely offset the initial weights. By themselves, the weights are without semantic content. Assume a hypothetical procurement where we only have two criteria to take into account. According to the EU directive 2004/17/EG on public procurement, we have to specify how the different criteria will be weighted when assessing the tenders. Assuming the weights would be

- Cost is 50 %
- Quality is 50 %

We receive two bids from suppliers, A and B. We create a score table as follows on a ten-point scale that we have defined in the specifications (Table 6.5)

Table 6.5 Evaluation of suppliers

	Cost	Quality
A	6	4
B	4	6

and obtain:

$$V(A) = 0.5 \cdot 6 + 0.5 \cdot 4 = 5$$
$$V(B) = 0.5 \cdot 4 + 0.5 \cdot 6 = 5$$

Now instead assume that when receiving the bids, we realize that we actually wanted another weighing, where quality should be more important than cost. In hindsight, we understand that what we really wanted was these weights:

- Cost is 25 %
- Quality is 75 %

This would give the following result:

$$V(A) = 0.25 \cdot 6 + 0.75 \cdot 4 = 4.5$$
$$V(B) = 0.25 \cdot 4 + 0.75 \cdot 6 = 5.5$$

However, we have already specified the weights in the tender documents for the legal requirement to be met, so we cannot alter those. But we can instead redefine the scales by calculating scaling factors. Assume that we have the weights w_i originally provided. Let v_i be our new weights and calculate $z_i = w_i/v_i$ (z_i are thus scaling factors for v_i). The scaling factors in our example are $25/50 = 0.5$ and $75/50 = 1.5$. Multiply the values with these and recalculate the mean values and keep the former weights (the law requirement is by this still fulfilled). And now we obtain (Table 6.6):

$$V(A) = 0.5 \cdot 3 + 0.5 \cdot 6 = 4.5$$
$$V(B) = 0.5 \cdot 2 + 0.5 \cdot 9 = 5.5$$

We simply adjusted the scale so that we obtain the desired result anyway, without changing the weights. The weights that we initially stated are preserved, but we shifted the scales so that they fully meet our new, revised preferences. Similarly, the order of the alternatives within each criterion is preserved. Naturally, this leaves a window for arbitrariness which by a skilled official can be used to circumvent the legal requirements of fairness, to award any winner he or she prefers. At first glance, the possibility to offset the weights completely by adjusting the value scales makes the entire weighing and assessing process meaningless.

Now, perhaps it can be argued that fairness and transparency should require us not to reset values in this way, and that the legal framework therefore should be modified

Table 6.6 Revised evaluation of suppliers

	Cost	Quality
A	3	6
B	2	9

as to prohibit this technique. One idea might be to require that the scales should be defined initially along with the weights, and that the procuring body should be required to stick to the initial scales. But the difficulty of attaching precise points to different options remains, particularly when we are dealing with qualitative values without an objective proxy measurement. And even if the procuring entity has the best possible intentions in terms of fairness and transparency, this type of problem can arise unnoticed when you simply use any kind of unreflected intuition.

The legal requirements are therefore insufficient in terms of how the valuation should be conducted. Many organizations have made extensive procurements according to the rules and in all respects in accordance with the EU directive on public procurement, or corresponding national legislation. The scales here therefore make no real sense as they are presently being used. But, as we will see below, this can be handled by an approach handling the imprecision in the situation in a proper way.

6.3 System Requirement Analysis

When devising an effective decision and evaluation process for public procurement, the three short-comings described above should be taken into account. First, when defining criteria importance, we should only state a ranking with some low level of qualification, to avoid unwarranted precision. In the example above, we could simply state that Cost is "considerably more important" than all other criteria, followed by Competence, which is "somewhat more important" than Responsiveness which in turn is "clearly more important" than the Design Concept. Secondly, when evaluating alternatives in different criteria, we should avoid assigning precise values, at least when there is no objective measure or proxy value at hand. Thirdly, the system should take the interdependence between weights and value scales into account.

To formalize the intuitive evaluation process, we should require a thorough needs analysis and extraction procedure. Such a procedure should additionally document what is in the decision-makers' heads. If the weights have been made public in the procurement documents, we need a framework for analysing each tender with respect to how they meet the criteria stated in the specifications. The proposed idea here is to start by ranking the alternatives in relation to each other under each criterion. Conceptually, this is simple. Either an alternative is as good as another, or it is better/worse. After having considered all the alternatives, we obtain a ranking from best to worst, possibly with several options as equally good at one or more places in the ranking. We now have a simple (ordinal) ranking.

The next step is to define the distances between the different alternatives, with respect to each criterion. Start with the best alternative and compare it with second best. Enter for each such pair whether the difference between them is small, medium or large. When this is done, we have a more qualified (cardinal) ranking. Continuing the example above, assume that our rankings look like the situation in Table 6.7.

Table 6.7 Evaluation of suppliers using an imprecise cardinal scale

Cost[a]	Competence	Responsiveness	Design concept
$A \succ\succ B$	$D \succ\succ B$	$C \succ B$	$D \succ A$
$B \succ\succ\succ C$	$B \succ C$	$B \succ\succ\succ D$	$A \succ\succ B$
$C \succ D$	$C \succ\succ\succ A$	$D \succ\succ A$	$B \succ\succ\succ C$

[a] $\succ$ represents 'a little better than', $\succ\succ$ 'clearly better than', and $\succ\succ\succ$ 'considerably better than'

Table 6.8 Criteria hierarchy

Cost $\succ\succ\succ$ Competence
Competence $\succ$ Responsiveness
Responsiveness $\succ\succ$ Design concept

Next, we indicate how the different criteria relate to each other in order to calibrate the scales. The approach is similar to what we did in order to rank the alternatives under each criterion above. The procedure is as follows:

1. Compare the criteria regarding their importance. Either a criterion has an equal importance as another one, or it is more or less important. After considering all the criteria, we get a ranking from most to least important, possibly with several criteria having equal importance in one or more places in the ranking. Thereafter we have a simple (ordinal) ranking of importance.
2. Enter the distances in the ranking. Start with the most important criterion and compare it with the second criterion in the ranking. Enter for each such pair of importance difference whether it is small, clear or significant. When this is done, we have a qualified (cardinal) hierarchy of importance.

Suppose we went through the above procedure with the ranking describe in Table 6.7 and obtained the qualified hierarchy in Table 6.8 between the potentials of the criteria in the example:

The notation used is similar to the notation for ranking alternatives, $\succ$ represents a 'small difference', $\succ\succ$ 'clear difference', and $\succ\succ\succ$ 'significant difference'.

From these comparisons, the cardinal number k_i for each criterion i can be calculated. A higher cardinal number indicates a higher potential of the criterion. The calculation of the cardinal number is quite straight-forward, and does not involve manually assigning any numerical values to the criteria potentials. Thereafter the scales are calibrated to correspond to what came up through the comparisons above, i.e., let k_i be the cardinal numbers and calculate $z_i = w_i/k_i$, where z_i are the calibration factors transforming the scale potentials to the predetermined weights w_i (e.g. stated in a procurement document). Thereafter, z_i are applied to the original values.

6.4 System Design

For a decision support system in the context outlined above we need calculations and a formal representation of the decision problem to be able to compute the best option and provide means for sensitivity analysis. Note that from the requirements, no numerical values are needed to be assigned any of the options under any criterion, however interval-values should be supported. In order to enable for decision evaluation with such prerequisites, a flora of different approaches have been suggested in the literature. Some approaches stems from the idea of representing imprecision in the form of intervals, or in more general terms, set-valued statements. In interval decision analysis, numerically imprecise information is modelled by means of constraints on variables for criteria weights and alternative values, see, e.g., [4, 19, 22, 27] for approaches relying on this interpretation. Other approaches aims to capture a set of value functions consistent with provided rankings [12, 14], models that exploit fuzzy numbers to represent numerical imprecision in rank statements [17] or representing linguistic statements such as "good" and "very good" using fuzzy numbers [13].

The approach presented below, called the Delta method from its primary decision evaluation rule [see Eq. (6.3)], is based upon constraint sets that complements range constraints (interval statements) with comparative statements. An important feature of the Delta method is the embedded sensitivity analysis and that rank statements can be mixed with interval statements, see [5, 8]. The computations become rather complex, involving maximization of non-linear objective functions such as the one in Eq. (6.1), but the computational mathematics is handled by platforms described in e.g. [3, 8, 9, 11, 18].

As mentioned above, at this stage in the procurement process, the problem to solve can be viewed as a multi-criteria decision problem. One large category of approaches to multi-criteria decision problems is where the decision criteria can be arranged in hierarchies, see Fig. 6.1.

For a criteria hierarchy, on each level the criteria are assigned weights and the alternatives are valued with respect to each sub-criterion. Flat criteria weight approaches can be seen as a special case—a one-level hierarchy. The minimum (or sometimes maximum) of the weighted value in Eq. (6.1) is usually employed as an evaluation rule.

For instance, in Fig. 6.1 above, the value of alternative A_i under sub-criterion jk is denoted by v_{ijk}. The weight of criteria j and sub-criteria jk are denoted by w_j and w_{jk} respectively, and the product term $w_{jk} \cdot v_{ijk}$ is referred to as the "part worth" value that A_i gets from criterion jk. The weighted, or aggregated, value of alternative A_i is

$$V(A_i) = \sum_{j=1}^{2} w_j \sum_{k=1}^{2} w_{jk} v_{ijk} \tag{6.1}$$

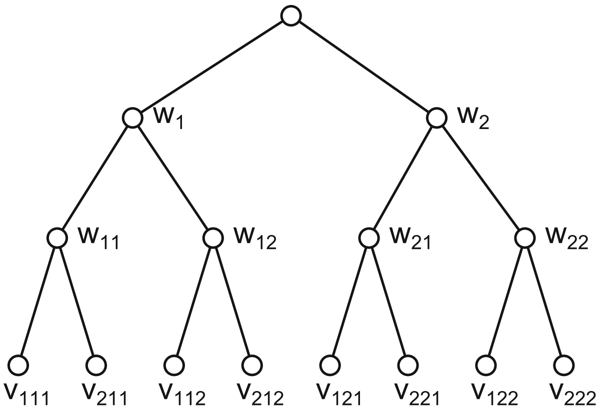


Fig. 6.1 A criteria hierarchy

Thus, if using a minimax approach, the alternative with the greatest minimum weighted value is suggested to be chosen. In this case of two alternatives evaluated under two criteria each having two sub-criteria, each weighted total value is a sum of four terms with each term containing three factors. A criteria tree is a symmetric tree. Given consequences c_i and c_j, denote their values v_i and v_j respectively. Then the user statements will be of the following kind for real numbers d_1, and d_2:

- Comparisons: v_i is from d_1 to d_2 larger than v_j, denoted $v_i - v_j \in (d_1, d_2)$ and translated into $v_i - v_j > d_1$ and $v_i - v_j < d_2$.

All the value statements in a decision problem share a common structure. Imprecise statements such as those in Table 6.7 are translated into comparisons which are constraints on the variables, and they are in turn translated into inequalities and collected together in a value constraint set. For weight statements, the same is done into a *weight constraint set*.

6.4.1 Node Constraint Set

The collection of value and weight statements in a decision situation is called the node constraint set. A constraint set is said to be consistent if it can be assigned at least one real number to each variable so that all inequalities are simultaneously satisfied.

The primary evaluation rule of the criteria tree is based on the hierarchical additive value function. Since neither weights nor values are fixed numbers, the evaluation of the hierarchical additive value yields multi-linear objective functions. Thus, given a criteria tree the hierarchical additive value of an alternative A_i, $HV(A_i)$, is given by

$$HV(A_i) = \sum_{i_1=1}^{n_{i_0}} w_{ii_1} \sum_{i_2=1}^{n_{i_1}} w_{ii_1 i_2 \ldots} \sum_{i_{m-1}=1}^{n_{i_{m-2}}} w_{ii_1 i_2 \ldots i_{m-2} i_{m-1}} \sum_{i_m=1}^{n_{i_{m-1}}} w_{ii_1 i_2 \ldots i_{m-2} i_{m-1} i_m} v_{ii_1 i_2 \ldots i_{m-2} i_{m-1} i_m} \tag{6.2}$$

where m is the depth of the hierarchy/tree corresponding to A_i, n_{i_k} is the number of child-criteria to a criterion with weight w_{i_k}, $w_{\ldots i_j \ldots}$, $j \in [1, \ldots, m]$, denote weight variables and $v_{\ldots i_j \ldots}$ denote value variables as above.

Optimisation with such non-linear expressions subject to linear constraints (the node and value constraint sets) are computationally demanding problems to solve for an interactive tool in the general case, using techniques from the area of non-linear programming. In the literature there are discussions about computational procedures to reduce non-linear problems to systems with linear objective functions, solvable with ordinary linear programming methods.

The area of linear programming (LP) deals with the maximizing (or minimizing) of a linear function with a large number of likewise linear constraints in the form of weak inequalities. Research efforts in the field are mainly focused on developing efficient representations and algorithms for finding local and global optima. The LP problem is the following optimising problem:

$$\begin{aligned} &\text{maximize } f(x) \\ &\text{when } Ax \geq b \\ &\text{and } x \geq 0 \end{aligned}$$

where $f(x)$ is a linear expression of the type $c_1x_1 + c_2x_2 + \ldots + c_nx_n$. $Ax \geq b$ is a matrix equation with rows $a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \geq b_1$ through $a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \geq bm$, and $x \geq 0$ are the non-negativity constraints $x_i \geq 0$ for each variable. Amongst all feasible points, the solution to $f(x)$ is sought that has the highest numerical value, i.e. the best solution vector x the components of which are all non-negative and satisfy all constraints. In the same way, a minimum can be searched for by negating all terms in the $f(x)$ expression. For the purposes of this system, $f(x)$ are expressions involving $HV(A_i)$ and $HV(A_j)$ for different alternatives A_i and A_j.

6.4.2 Comparing Alternatives

Alternatives are compared according to their (hierarchical) additive value. However, for evaluation purposes the notion of strength is introduced, where the strength of A_i to A_j is simply the difference

$$\delta_{ij} = HV(A_i) - HV(A_j) \tag{6.3}$$

where $\delta_{ij} > 0$ would mean that A_i is preferred to A_j. However, if $HV(A_i)$ and $HV(A_j)$ are interval-valued, δ_{ij} is interval-valued as well which may lead to overlapping value intervals meaning that preference is not that straightforward to conclude. To handle this, the concept of contraction has been proposed as an embedded form of sensitivity analysis when dealing with overlapping value intervals.

The contraction analysis consists of (proportionally) shrinking the range of each weight and value interval while studying $\max(\delta_{ij})$ and $\min(\delta_{ij})$ at different contraction levels. This means that the optimisation problem becomes obtaining $\max(\delta_{ij})$. The level of contraction is indicated as a percentage, so that for a 50 % level of contraction the range of each variable (weight, value) interval has been reduced to half their initial length and for a 100 % level of contraction to the marginal centroid value. Hence, contraction is done towards each polytope centroid given an (implicitly stated) uniform second-order distribution over each solution set, see, e.g., [8]. This means that a traditional point-wise result (coinciding with the 100 % contraction level) can be analysed in all dimensions simultaneously by widening the intervals from a point (interval with zero width) to for example 50 % contraction (halfway between the point and the maximum intervals consistent with the input information). The underlying assumption behind the contraction analysis is that the there is less belief in the outer endpoints of the intervals than in points closer to the most-likely point, which also has been verified in, e.g., [24]. The level of contraction required in order to get δ_{ij} being always positive or always negative is then used as an embedded form of sensitivity analysis, used to inform on the confidence in the resulting ranking of alternatives.

6.5 System Development

The starting point of the system development was the identification of procurement vendor selection processes with general MCDM structures and solution methods. The selection of one or more vendors from a set of offers in a procurement process can be seen as equivalent to the problem of selection one or more alternative courses of action from a predetermined set. The latter situation is a traditional multi-criteria problem, with each vendor mapped onto one course of action and the criteria that determines the best offer are of the same nature as ordinary criteria in MCDM. The importance of such criteria can, in procurement as well as in general MCDM, be expressed as (explicit or implicit) weights. By this identification, procurement evaluation processes can be mapped onto MCDM structures and that was the realisation that led to the development of the system in this paper.

The task of designing and developing software containing complex algorithms that are not easy to imagine and completely specify beforehand requires some specific approach regarding choice of design methods. In this article, we discuss the development of algorithms originally intended for decision analysis [6] and now adapted for procurement evaluation. The moving target nature of developing this system required different development techniques and approaches than more ordinary software development efforts would require.

Developing software containing complex algorithms differs from everyday software development in some respects. In most software development, the design can be planned in an orderly fashion using experience or extrapolation from previous development projects. In many cases, parts of the code can even be reused or at least patterns of design can be reused. But in designing algorithm centred software, containing new algorithms or new requirements unknown at specification time, what is normally good software design practices cannot always be applied or would not lead to effective development work.

For example, while object-oriented design and coding is often good practice, it might become a hindrance when there are no natural objects to discover or structures cannot be manipulated in detail in an implementation independent way. This article describes the use in procurement evaluation of a software library for decision analysis that was developed for maximal efficiency and minimal footprint. At its core is a linear optimisation algorithm but not of the traditional kind trying to solve a very large problem using minutes of CPU time or more. This library is the basis for an interactive procurement tool and the response needs to be experienced as immediate. Also, there is not one but a set of LP problems to be solved in sequence in order to obtain the sensitivity analysis (contraction). The sequence of LP problems solved are correlated and most of the effort expended is used to find, keep track of, and exploit the similarities of the problems in order to minimise the solution times.

6.6 User Interface Design

The user interface should enable for a user with little or no previous experience of multi-criteria decision analysis to provide input statements conforming to the Delta method. Two different user interfaces exists, one consisting of the *Decide*IT decision tool implemented in the Java programming language, and one implemented as a web based tool in Node.js called the “Preference Decision Wizard”. Both tools employs a step wise process for modelling and evaluation of multi-criteria decision problems, and designed in collaboration with the Stockholm University procurement unit. The step wise process follows the typical decision analysis process. The process holds with the following steps:

Step 1—Set suppliers
The set of suppliers whose tenders fulfil the basic requirements is added. These are the alternatives of the decision model.

Step 2—Set main criteria and optionally also sub-criteria
The set of evaluation criteria is added. Some criteria may be further divided into sub-criteria. See Fig. 6.2.

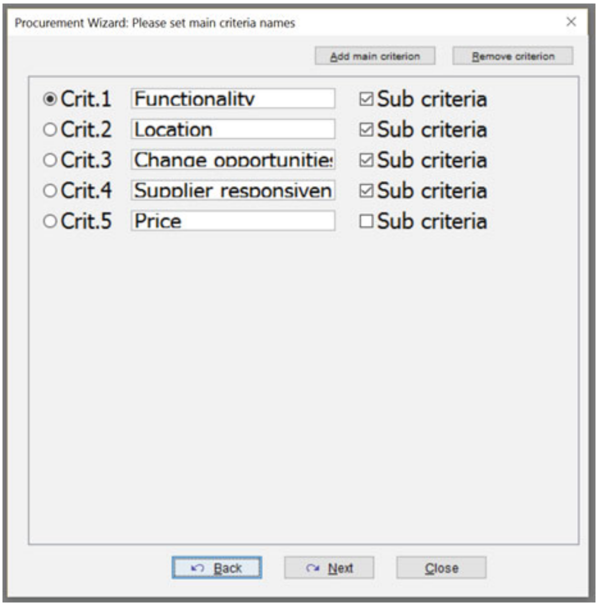


Fig. 6.2 Entering main criteria

Step 3—Value/rank suppliers
A cardinal ranking of the suppliers is done for each criterion. In the case of sub-criteria, this is done for each sub-criterion. See Fig. 6.3.
Step 4—Weigh/rank criteria
A cardinal ranking of the criteria is done, both for the main criteria and the sub-criteria belonging to the same main criterion.
Step 5—Evaluate suppliers
Decision evaluation of the suppliers are done. Essentially, all evaluation methods of the *Decide*IT tool is available, although the procurement module propose a default evaluation presentation format in the form of part-worth bar charts, showing the contribution from each criterion in a bar chart where the contribution is obtained from the product of the weight and value centroids.

6.7 System Implementation

The computational library package was initially designed using a contract based specification, using an object-based approach (where object-based refers to object-oriented minus inheritance). The use of non-inheriting objects led to a design that could survive changing requirements over time, while at the same time not enough natural code objects were found to allow an efficient implementation using

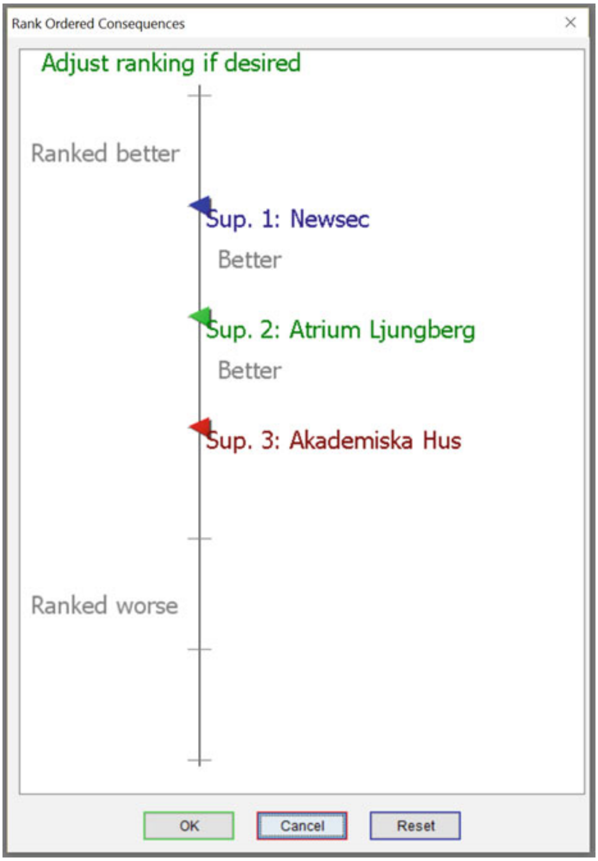


Fig. 6.3 Ranking of suppliers

object-oriented programming. Issues of code optimisation and footprint minimization were handled by using a pure imperative language without object extensions, in this case C.

Using conditional compilation and macros, techniques akin to aspect-orientation was used in coding parts particularly involving memory management, logging, and exception handling. The main challenge has been algorithmic complexity and changing requirements and specifications. One solution to the changing requirements problem is a configuration program on a meta level. The source code is then automatically recompiled prior to execution on a new platform. In this way, the source code becomes optimised for the actual target machine. Such inclusions are simpler to manage using aspect-structured code than object-oriented code since different hardware influenced aspects are more important than an object hierarchy would have been in determining the success of the software.

This is not to argue that object-orientation is less usable in general for developing calculation intense algorithms. But the point here is that code objects are not always the primary code structure choice, especially not when the outcome of the coding effort is partly unknown at the outset. The concept of aspect-oriented software design deals with architectural crosscutting concerns such as, i.a., memory management, code optimisation, and real-time behaviour. The software package was designed using an object-based overall design approach but was implemented using coding techniques more closely related to aspect-oriented techniques.

For integration purposes with the user interfaces, for *Decide*IT, the computational C library is wrapped in a C++ layer using the Java Native Interface, for the Preference Decision Wizard a C++ wrapping layer together with a C# API is implemented, using DLLImport interoperability.

6.8 System User Experience

We describe a simple example of how a more adequate support can easily manage a procurement situation and obtain a significant increase in quality in the assessment stage. The example is from Stockholm University's procurement of new facilities at Campus Kista. It was decided, in brief, to obtain, for one of the university's departments, new premises since the existing ones had become inadequate. This was a 90 million EUR investment. The criteria emphasized the premises *functionality*. Furthermore they emphasized *localization* (which implicitly was within Kista Science City) and opportunities for *interaction with the surrounding society* and the *possibilities for change*, *flexibility* and the supplier *responsiveness and innovation levels*. To these main criteria, sub-criteria were added under the relevant criteria. Finally the price parameter was asserted. Note in particular that the price criterion was in no way perceived as decisive. The main and sub-criteria were obviously mainly of a qualitative nature.

Following the announcement, it was found that there were three facility suppliers who met the criteria: Newsec, Atrium Ljungberg, and Akademiska hus. Thereafter, the analysis began and preferences could be established within the evaluation team. The notation in the tables below expresses the evaluation team preferences. For example, the ordering for criterion 1 (Symbolic value) is $a2 \succ a1 \succ a3$, meaning that it was perceived by the group that Atrium Ljungberg was the vendor that best met the criterion, followed by Newsec and then Akademiska hus. The notation $\{a1, a2, a3\}$ means that the suppliers perceived to be equivalent under the criterion. Analogously, $4 \succ 1 \succ \{2, 3, 5, 6\}$ means that criterion 4 (Realization of office space in the building) were perceived as more important than symbolic value, which in turn was seen as more important than the criteria 2, 3, 5 and 6. These latter were perceived as equivalent. The categories were prioritized in the order they are presented in Table 6.9. The most important group of criteria was the ones listed under Functionality, the second most important group was the ones listed under Location, etc. Note that the evaluation team chose to work exclusively qualitatively,

Table 6.9 Evaluation team preferences

Category	Criterion	Ranking[a]
Functionality	1. Symbolic value	$a2 \succ a1 \succ a3$
	2. Contributions to social contacts	$\{a1, a2\} \succ a3$
	3. Access to public facilities	$a2 \succ \{a1, a3\}$
	4. Realization of office space	$\{a1, a2, a3\}$
	5. Technical standard	$\{a1, a2, a3\}$
	6. Environmental requirements	$\{a1, a2, a3\}$
		$4 \succ 1 \succ \{2, 3, 5, 6\}$
Location	7. Interaction with the environment	$a2 \succ a3 \succ a1$
	8. Access to common facilities	$a2 \succ \{a1, a3\}$
		$7 \succ 8$
Change opportunities	9. Interaction during planning phase	$\{a1, a2, a3\}$
	10. Interaction during contract phase	$\{a2, a3\} \succ a1$
	11. Flexibility during planning phase	$\{a1, a2, a3\}$
	12. Flexibility during contract phase	$\{a1, a2, a3\}$
		$\{9, 10\} \succ \{11, 12\}$
Supplier responsiveness	13. Responsiveness	$\{a1, a2, a3\}$
	14. Innovation	$a2 \succ \{a1, a3\}$
		$14 \succ 13$
Price	15. Price	$\{a3, a2, a1\}$

[a] a1 = Newsec, a2 = Atrium Ljungberg, a3 = Akademiska hus

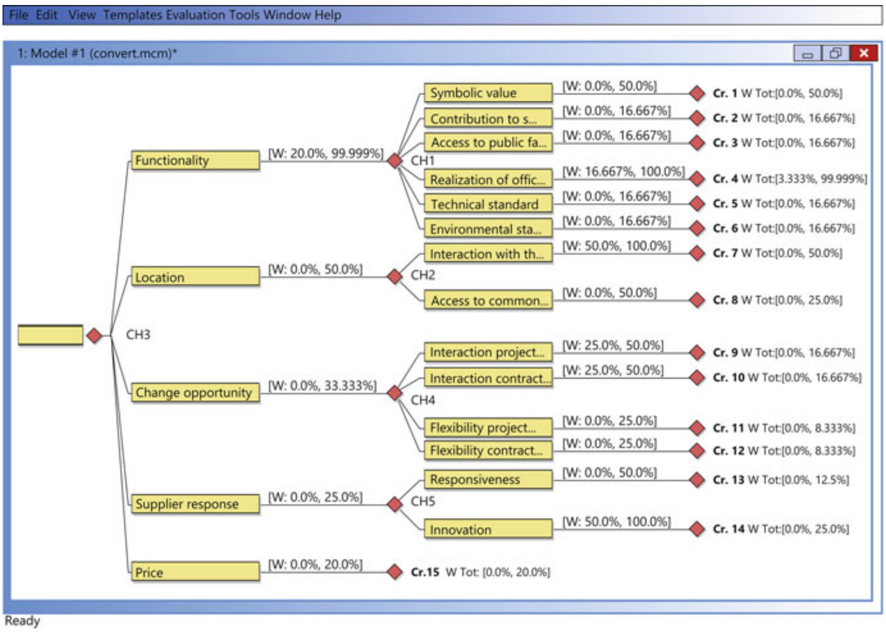


Fig. 6.4 Multi-criteria decision tree generated

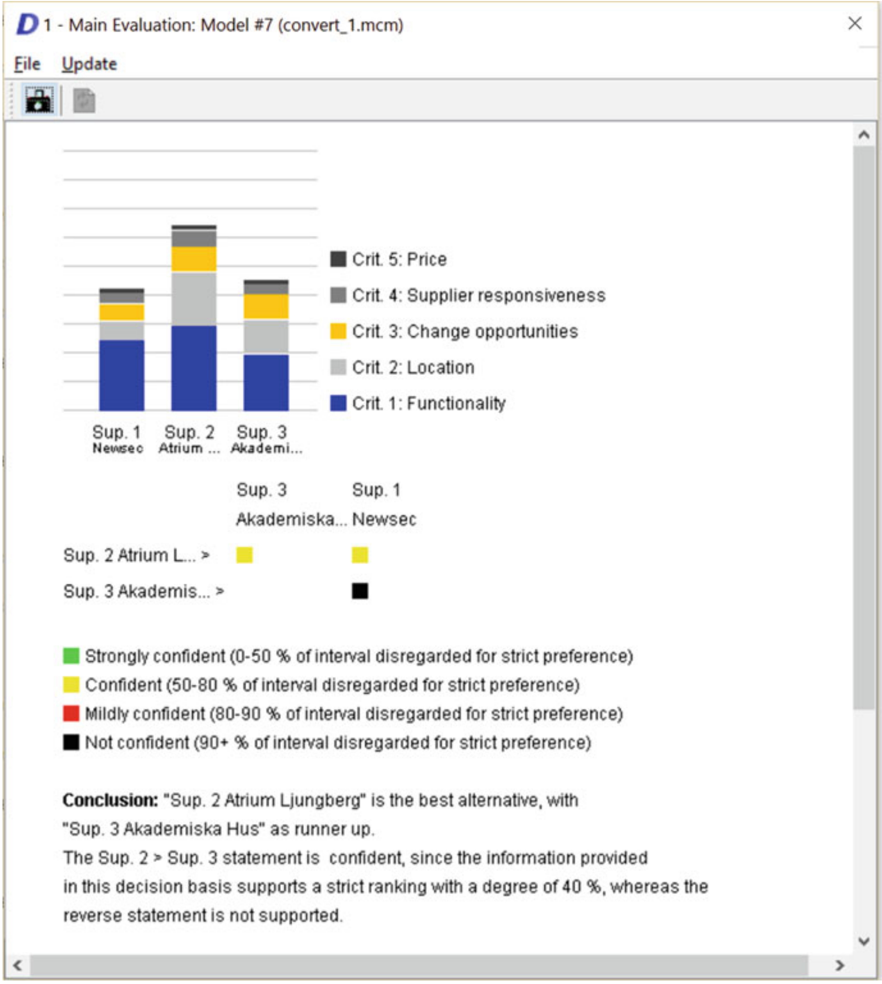


Fig. 6.5 Main evaluation window with main criteria part worth bar charts for each supplier. The level of contraction level needed in order for δ_{23} to be always positive is 40 %, here calibrated to indicate a "confident" ranking

but there are no computational obstacles to introduce range or precise figures in this analysis and mix with qualitative statements.

Using this information as input to the decision support system entered through the step wise process, a multi-criteria decision can be generated if using the *Decide*IT tool, as shown in Fig. 6.4.

Then the decision problem was evaluated, below the final result of the main evaluation window is shown, see Fig. 6.5.

The final result is not a numerical value, but something richer. The methods used managed to preserve some level of imprecision, reflecting the qualitative nature of the input, without making sacrifices in terms of transparency or deterministic output. In short, the figure above says that Atrium Ljungberg is the supplier that best met the criteria, followed by Newsec and Akademiska hus. The latter were reasonably equivalent alternatives.

6.9 Concluding Remarks

We have demonstrated three fundamental problems with the models normally used in procurement situations. These are so serious so that procurements often lose their meaning and lead to large and costly miscalculations. We have shown how by managing them systematically the quality of analysis can significantly improve and, despite the use intuitively more natural assessments of suppliers' proposals, we can get a result that gives a fuller analysis. The method not only points out the supplier who should be awarded the contract. If there is a candidate that is better than the other as it gives a much clearer picture of the situation and pointing out where analysis critical points. Note here that this cannot be done without the rather elaborate nonlinear optimisation algorithms, but a number of specially developed those found in tool *Decide*IT used in the analysis. By utilizing this tool qualitative statements could therefore be easily managed without the need to introduce artificial and somewhat arbitrary conversion formulas.

In conclusion we note that purchasers clearly need support tools beyond those that only ascertain that the formal requirements are met, i.e. there is a need for tools that are using appropriate calculation functions that support scalable management and allow imprecision in value and weight statements. These features can then be easily integrated into a complete support tools to support both the formal process calculation steps. *Decide*IT is one such tool that fairly easily and properly handle mixtures of qualitative and quantitative criteria.

The algorithms for solving the evaluations of cardinal rankings are optimisation algorithms, but of a slightly different nature than ordinary optimisation problems. In ordinary optimisation, the task is often to find a local optimum (sometimes a global) for a problem with many variables, possibly millions. This is often done in batch mode, i.e. the real-time (or interactive) requirements are low. But in this case, the design is required to solve many (hundreds) of optimisation problems in fractions of a second, the speed requirement being that the user should not experience any delay in response. For this to be possible, a network of result caches had to be devised. While the exact design of the caches are not important, it is interesting to note that these kinds of requirements are not easily anticipated before the specific procedure was produced. Thus, the overall software design depends on algorithms whose specifications are not known from the outset and whose development cannot be foreseen since there are no originals or templates to start with.

In summary, the library package is still alive and continues to evolve at the research frontier, more than 15 years after its first release without requiring a rewrite or architectural redesign. Part of its longevity, despite complexity and changing requirements and specifications, is due to the following set of principles:

- an object-based approach
- a contract based specification
- aspect-orientation-like management of key code features
- a pure imperative programming language

resulting in reasonable development control without introducing overhead in the form of over-specification, slow execution, or a too large footprint.

References

1. Bergman, M.A., Lundberg, S.: Tender evaluation and supplier selection methods in public procurement. J. Purch. Supply Manag. **19**(2), 73–83 (2013)
2. de Boer, L., Labro, E., Morlacchi, P.: A review of methods supporting supplier selection. Eur. J. Purch. Supply Manag. **7**(2), 75–89 (2001)
3. Caster, O., Ekenberg, L.: Combining Second-Order Belief Distributions with Qualitative Statements in Decision Analysis, pp. 67–87. Springer, Berlin (2012)
4. Danielson, M.: Handling imperfect user statements in real-life decision analysis. Int. J. Inf. Technol. Decis. Mak. **3**, 513–534 (2004)
5. Danielson, M.: Sensitivity analysis in interval decision modelling. Eng. Lett. **17**(1), 1–7 (2009)
6. Danielson, M., Ekenberg, L.: A method for decision and risk analysis in probabilistic and multiple criteria situations. Patent (2007). U.S. Patent No. 11/145229
7. Danielson, M., Ekenberg, L., Johansson, J., Larsson, A.: The decideit decision tool. In: ISIPTA 03, pp. 204–217 (2003)
8. Danielson, M., Ekenberg, L., Idefeldt, J., Larsson, A.: Using a software tool for public decision analysis: the case of Nacka municipality. Decis. Anal. **4**, 76–90 (2007)
9. Danielson, M., Ekenberg, L., Ekenberg, A., Torsten Hökby, Lidén, J.: Decision process support for participatory democracy. J. Multi-Criteria Decis. Anal. **15**, 15–30 (2008)
10. Danielson, M., Ekenberg, L., Larsson, A., Riabacke, M.: Transparent Public Decision Making: Discussion and Case Study in Sweden, pp. 263–281. Springer, Berlin (2010)
11. Ding, X.S., Danielson, M., Ekenberg, L.: Non-linear programming solvers for decision analysis support systems. In: International Conference on Operations Research 2003, pp. 475–482 (2004)
12. Figueira, J.R., Greco, S., Slowinski, R.: Building a set of additive value functions representing a reference preorder and intensities of preference: grip method. Eur. J. Oper. Res. **195**(2), 460–486 (2009)
13. García-Cascales, M.S., Lamata, M.T.: Solving a decision problem with linguistic information. Pattern Recogn. Lett. **28**, 2284–2294 (2007)
14. Greco, S., Mousseau, V., Slowinski, R.: Ordinal regression revisited: multiple criteria ranking using a set of additive value functions. Eur. J. Oper. Res. **191**(2), 416–436 (2008)
15. Harland, C., Telgen, J., Callender, G.: International Research Study of Public Procurement, pp. 372–399. SAGE, London (2013)
16. Ho, W., Xu, X., Dey, P.K.: Multi-criteria decision making approaches for supplier evaluation and selection: a literature review. Eur. J. Oper. Res. **202**(1), 16–24 (2010)
17. Jiménez, A., Mateos, A., Sabio, P.: Dominance intensity measure within fuzzy weight oriented MAUT: an application. Omega **41**(2), 397–405 (2013)
18. Larsson, A., Johansson, J., Ekenberg, L., Danielson, M.: Decision analysis with multiple objectives in a framework for evaluating imprecision. Int. J. Uncertain. Fuzz. Knowl. Based Syst. **13**(5), 495–509 (2005)

19. Lee, K.S., Park, K.S., Kim, S.H.: Dominance, potential optimality, imprecise information, and hierarchical structure in multi-criteria analysis. Comput. Oper. Res. **29**(9), 1267–1281 (2002)
20. Lunander, A., Arne, A.: Metoder vid utvärdering av pris och kvalitet i offentlig upphandling, 11 (2004)
21. McCrudden, C.: Using public procurement to achieve social outcomes. Nat. Res. Forum **28**(4), 257–267 (2004)
22. Salo, A.A., Hämäläinen, R.P.: Preference ratios in multiattribute evaluation (prime) – elicitation and decision procedures under incomplete information. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. **31**(6), 533–545 (2001)
23. Stilger, P.S., Siderius, J., van Raaij, E.M.: A Comparative Study of Formulas for Choosing the Economically Most Advantageous Tender (2015). http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2626934
24. Sundgren, D., Danielson, M., Ekenberg, L.: Warp effects on calculating interval probabilities. Int. J. Approx. Reason. **50**, 1360–1368 (2009)
25. Telgen, J., Uenk, N., Lohmann, W., Manunza, E.: Conditions on lots: practical desirability versus legal acceptability. In: Piga, G., Tatrai, T., (eds.) Public Procurement Policy. The Economics of Legal Relationships, pp. 90–98. Routledge, London (2015)
26. Weber, C.A., Current, J.R., Benton, W.C.: Vendor selection criteria and methods. Eur. J. Oper. Res. **50**, 2–18 (1991)
27. Weber, M.: A method of multi attribute decision making with incomplete information. Manag. Sci. **31**(11), 1365–1371 (1985)

Chapter 7
Evaluation Support System for Open Challenges on Earth Observation Topics

António J. Falcão, Rita A. Ribeiro, Javad Jassbi, Samantha Lavender, Enguerran Boissier, and Fabrice Brito

Abstract This chapter presents a model-driven evaluation support system (ESS) for open competitions within Earth Observation (EO) topics, which was part of a project financed by ESA (European Space Agency). One objective of the project was to deliver a collaborative platform that, through Data Challenges (i.e. open competitions), could improve the adoption and outreach of new applications and methods to process EO data. A second objective was to develop a common environment (web portal) where the challenges could be developed, deployed and executed; and thirdly, to easily publish results in a visualisation platform providing a user friendly interface for validation, evaluation and transparent peer comparisons. In general, the developed tool is an evaluation support system with four main requirements: support initiators/evaluators to define a general challenge in the EO domain; enable easy interaction with the candidates/contestants to submit their entries; provide criteria ratings and candidate evaluation score; and transparent peer comparison. Further, the ESS should be adaptable to several different topics within the EO domain and robust enough to deal with qualitative and quantitative data. In this work the focus is on the conceptual evaluation model of ESS, which is based on multi-criteria concepts and a data fusion algorithm capable of standardizing all criteria and providing a sophisticated aggregation process to rank the candidates. The proposed ESS model includes a tree representation of the criteria and their relationships, and then a bottom-up hierarchical process to aggregate the results from the leaf nodes to the top final score. To better explain the evaluation model

A.J. Falcão (✉) • R.A. Ribeiro • J. Jassbi
UNINOVA, Campus FCT/UNL, Monte da Caparica 2829-516, Portugal
e-mail: ajf@uninova.pt; rar@uninova.pt; j.jassbi@uninova.pt

S. Lavender
Pixalytics Ltd, 1 Davy Road, Tamar Science Park, Derriford, Plymouth, Devon PL6 8BX, UK
e-mail: slavender@pixalytics.com

E. Boissier • F. Brito
Terradue UK Ltd, 11 Old Jewry, London EC2R 8DU, UK
e-mail: enguerran.boissier@terradue.com; fabrice.brito@terradue.com

© Springer International Publishing Switzerland 2016

J. Papathanasiou et al. (eds.), *Real-World Decision Support Systems*,
Integrated Series in Information Systems 37,
DOI 10.1007/978-3-319-43916-7_7

and the bottom-up synthetizing process we used a step-by-step example and then an experimental case.

7.1 Introduction

In this chapter we present a decision model for an evaluation support tool to rank candidates/contestants of open challenges within Earth Observation (EO) domains. This support system can be classified as a model-driven DSS, according to the classification in Chap. 1. This tool was developed in the context of the E-CEO project [5], financed by ESA (European Space Agency), and included two case studies (data challenges), one devoted to Ocean Colour Atmospheric Correction [11] and another to Ice Velocity of the Antarctic [6]. These two challenges already produced results and were run on the actual Web portal at ESA [6]. Further, the project leader, TerraDue, implemented the software tool and UNINOVA team devised the conceptual decision model described in this chapter. The main objectives of the developed evaluation support system (ESS) were: (1) to deliver a collaborative platform that, through different data challenges, i.e. open contests on EO topics, could improve the adoption and outreach of new applications and methods to process EO data; (2) to develop a common environment where the challenges could be developed, deployed and executed; (3) and where the candidate results would be easily published in a common visualization platform for their effective validation, evaluation and transparent peer comparisons.

The focus on this chapter is to discuss the envisaged conceptual model of the ESS. The challenges for developing this model were: (a) to define a general evaluation tool in the EO domain to deal with different types of topics (e.g. remote sensing interferometry, ocean colour atmospheric corrections, ice sheets melting velocity); (b) enabling heterogeneous types of data (e.g. statistical data, expert judgments, computing performance); (c) enable cross-comparison between candidates to make the evaluation process transparent; (d) ensuring the synthetizing process and algorithms would work autonomously and robustly to enable ranking candidates from very different challenges. Furthermore, we also took in consideration that both the model and measures used were comprehensible enough for non-experts on decision making. To solve those challenges, we proposed an approach involving a multi-criteria decision making (MCDM) paradigm, which proved to be particularly useful for both structuring the competition problem (open contests) and to determine the candidates' evaluations/scores, in an understandable and structured framework. In addition, we borrowed parts of a data fusion process developed by our group [17] for our MCDM model, such as the fuzzy normalization process (to allow comparable and numerical input data) and the weighting aggregation operator for determining the alternatives rating. Summarizing, a sophisticated decision support system capable of supporting the initiators/evaluator to define EO competitions and evaluating the candidates in a straightforward way, allowing transparent peer comparisons, was defined for this project.

This first section introduces the scope and motivation for this work and presents an overview of the decision model paradigm as well as the background concepts and technologies behind the evaluation model. Section 7.2 presents the ESS design, while Sect. 7.3 discusses the requirements analysis and steps of the quantitative and qualitative evaluation model. Section 7.4 provides an overview of the user interface design. Section 7.5 then discusses the system implementation, detailing the proposed bottom-up hierarchical process for obtaining the ratings, i.e., the aggregation process to obtain the final candidates' scores. Section 7.6 presents, first, a small illustrative example to explain step-by-step the proposed model and, second, one experimental case to demonstrate the versatility of the implemented model and how it enables transparent peer comparisons. Finally, Sect. 7.7 summarizes the evaluation decision model defined in this work.

Background Concepts and Technologies Used in the ESS Decision Model Decision making may be characterized as a process of choosing or selecting a 'good' alternative(s), from a set of alternatives, to attain a goal or goals [14, 19]. In general, the computerized process of decision making can be divided in three main phases: intelligence, design and choice [20]. Figure 7.1 describes the various steps of each phase from the initial problem definition to the implementation of a solution.

In general, evaluation of alternatives using a set of criteria is the goal of Multiple Criteria Decision Making (MCDM) methods or techniques [19, 22]. Their objective (crisp or fuzzy) is to support decision makers/stakeholders by providing a systematic

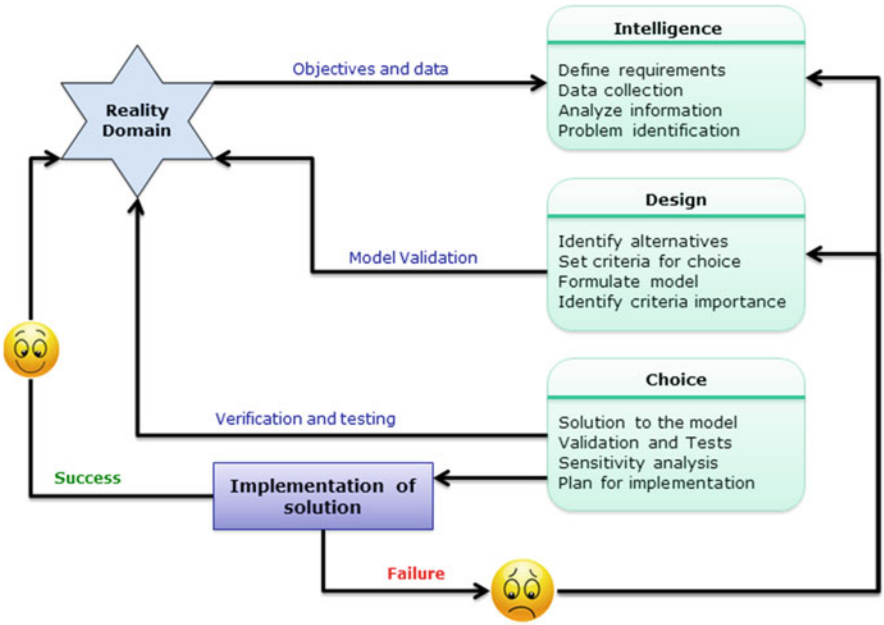


Fig. 7.1 Decision making process (adapted from Turban and Aronson [20])

process for prioritizing a set of alternatives (e.g. output results) based on evaluating a set of criteria [19]; this area is also sometimes called Multiple Attribute Decision Making [22]. This paradigm is the basis for this work because we are discussing an evaluation decision model for assessing contests (challenges) on real EO projects. In general, this paradigm requires:

- Definition of the problem structure of criteria and their hierarchical dependencies (e.g. decision tree);
- Criteria must be numerically comparable to be combined (normalized);
- Definition of relative importance of criteria (usually by experts);
- Selection of aggregation operator to obtain the alternative's rating.

Criteria can be either assessed with qualitative scores (e.g. "very good") or with quantitative scores (e.g. "% of CPU usage"). To be able to aggregate the various scores—obtained by each alternative regarding each criterion—data must be numeric and comparable, as stated in the second bullet above (2). This process is usually called normalization and can be achieved by using different normalization techniques, ranging from simple linear ones (e.g. division by maximum value of all alternatives for a specific criterion) to more sophisticated ones, such as functions representing the "goodness" of the criteria values (also called fuzzification [18]); or cross-comparison methods, e.g. normalization by comparing with max and min values obtained by all alternatives for the criteria [10].

The relative importance of each attribute, bullet (3) above, requires the specification of a preference structure over a set of attributes, to reflect the relative importance of each attribute in the composite. Procedures for eliciting and determining relative importance (in terms of weights) have been the focus of extensive research and discussion; see for example [3, 9, 13, 15]. There are three main approaches in order to choose weights [13]: indifference trade-off method, direct weighting methods, and probabilistic equivalence. Indifference trade-off methods address the question of providing a measure of how much the expert is willing to give up in one attribute to obtain improvements in another one; weights are calculated by the given indifference judgments. Direct weighting methods are the most common for determining weights, because they rely mainly on expert's know-how. Examples of direct weighting methods are: rank ordering and the Analytical Hierarchy Process (AHP) method [19, 22]. In classical approaches, such as decision trees and expected value approaches, the most common weighting method is to assign probabilities of occurrence of events and then use a weighted average for determining the ratings [19].

The classical mathematical formulation of multiple criteria decision making (MCDM) problems to determine the rating of an alternative—i.e. aggregation of criteria scores to obtain the rating—is:

$$D\left(A_i\right) = \bigoplus_i \left(w_{ij} \bigotimes x_i\right) \tag{7.1}$$

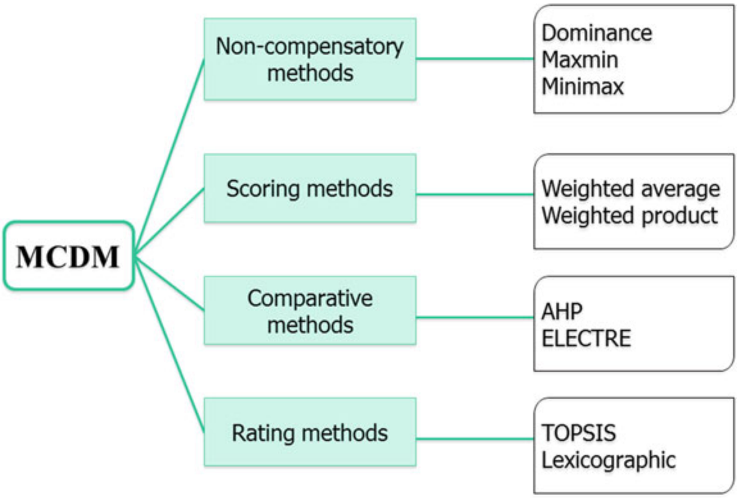


Fig. 7.2 Taxonomy for aggregation methods [16]

where w_{ij} is the relative importance (weight) of attribute x_j and $\bigoplus$, $\bigotimes$ are aggregation operators. The best alternative A_i is the one with the highest priority in the ranking.

There are many aggregation methods/operators—bullet (4) above—proposed in the literature for obtaining the rating of alternatives [1, 2, 21, 22]: scoring and outranking methods, trade-off schemes, distance based methods, value and utility functions, interactive methods. Figure 7.2 depicts a simple taxonomy for guiding the selection of methods/techniques with well-known examples [16].

Scoring methods are widely used, particularly the weighted average/mean method (classical method above using arithmetic operators), which is based on the multiple attribute value theory [21]. It is also worth mentioning that within the scoring class there is a large panoply of additive operators, to be used for obtaining the final rating of each alternative; ranging from mixture operators to full reinforcement operators [1]. In this work, we used a combination of a simple average for the statistical measures with mixture operators with weighting functions [13, 15] for the non-statistical criteria. The latter mixture operators with weighting functions, part of the data fusion process [17] ensure that criteria relative importance is penalized or rewarded, depending on their level of achievement.

There are interesting proposals in the literature for evaluation applications (see for example [2, 4, 7, 8, 12]), as well as many software tools implementing different multi-criteria decision making methods, both academic and commercial. However, for our purposes none of the tools could satisfy all required challenges of the case study therefore we built a specific evaluation model using a hybrid approach.

In summary, multi-criteria analysis appeared in the 1960s as a decision-making technique. It is mostly used to make a comparative assessment of candidate alternatives (e.g. services, software components, projects etc.) and to rank them. MCDM paradigms are designed to help decision-makers to integrate different options, reflecting opinions of actors concerned, into a prospective or retrospective framework. Multi-criteria solving methods are organized with the aim of producing a single synthetic conclusion at the end of the evaluation or, on the contrary, with a view of producing conclusions adapted to the preferences and priorities of several different partners.

7.2 Evaluation Support System (ESS) Design

To define the decision model for the evaluation support system (ESS), three main phases of the decision making process (Fig. 7.1) were considered, as follows.

Intelligence Phase

In the intelligence phase, the main challenge was to identify the problem and define how many and which criteria would be considered in Earth Observation (EO) contexts [5]; any defined challenge would require that candidates submit novel software algorithms for solving a particular EO problem. Therefore, it was important to identify dependencies between potential criteria for enabling construction of the complex hierarchical criteria evaluation decision tree; to this aim a questionnaire was presented to three domain experts [5]—partners in the project—to extract the most important factors to consider. In addition to the questionnaire, several email exchanges and teleconferences were held to reach a consensus about the hierarchical evaluation criteria tree.

The hierarchical criteria organization took into consideration three main goals, implicit in any EO challenge: efficiency, effectiveness and quality of the candidate's submissions. In the EO domain the efficiency criteria will measure the software performance; the effectiveness will measure the candidate's algorithm correctness; and the quality will assess the novelty of the approach. Each criterion is further divided into sub-criteria, in several layers, that represent the sub-criteria dependencies. We opted for representing all criteria as a hierarchical tree, to organize and structure a complex problem as well as clarify the dependencies and relationships between criteria.

Also, in this phase, it was selected which MCDM method would be used in the challenges evaluation. Since both qualitative and quantitative criteria need to be considered and there are dependent sub-criteria, we used a hierarchical decision tree solved with a synthetizing data fusion process using mixture operators with weighting functions [17]. The rational for our choice was that the evaluation support system (ESS) had to be simple and understandable by evaluators and domain experts, without failing to take into account generality, intrinsic uncertainties in the challenges and qualitative judgments from EO evaluators.

Design Phase

In this phase, the criteria evaluation tree was constructed with the respective layers of sub-criteria. The focus was on the leaf nodes because these need to be implemented in the ESS framework to obtain the respective scores. The upper layers of each branch are evaluated by means of the proposed hierarchical synthetizing process, as described in Sect. 7.5.

The devised sub-criteria definition was a complex task due to the diverse nature of the criteria types, from qualitative, statistical, cross-comparison and so forth. However, their standardized definition allowed a straightforward implementation within the framework, except for the qualitative criteria, which required a preliminary step, where the evaluator (or challenge initiator) is asked to evaluate (in a simple semantic form) the sub-criteria involved. Since there are only three qualitative sub-criteria, it is a small overload for an evaluator and the respective outcome will be aggregated with the quantitative classifications. It is important to highlight that the implemented framework calculates the values for each leaf node measure (quantification column) and its respective score (score implementation). For the cross-comparison criteria the framework needs to first calculate the scores for all contestants, to retrieve the max (best) and min (worst), which will then be used in the respective cross-comparison normalization (details in Sect. 7.3.2).

The definitions for each sub-criterion, exemplified in Table 7.1, include: (a) objective of the measure; (b) unit and dimension; (c) quantification/formulation; (d) implicit logic of the criterion; (e) normalization process; (f) actor (responsible entity for calculation). A tabular form was used to simplify and standardize the descriptions, as shown in Table 7.1.

Further, the method for determining the relative importance for the criteria is based on weighting functions [13, 15], which dynamically adjust the importance/weights assigned to each criteria, as a function of the degree to which each is being satisfied, i.e. increasing or decreasing the weight according to the satisfaction level of the criterion.

Choice Phase

In this phase a hierarchical synthetizing process was devised, which includes two steps: (a) a simple averaging aggregation for the leaf nodes of the quantitative and statistical measures; (b) an aggregation process using mixture operators with weighting functions [8, 13] for the upper layers. The whole process is composed of a bottom-up synthetization to obtain the final score for each challenge candidate.

A validation and testing campaign was planned and executed. This included a first validation with one experimental challenge case prepared by a domain expert and validated in a spreadsheet application; an adapted version of this case is used for illustrative purposes in this chapter. After the first validation, a similar case study—also prepared by the same domain expert—was used to test and validate the implemented framework.

7.3 Evaluation Support System Requirements Analysis

From the intelligence phase, described in the previous section, we identified three main challenges for assessing any EO challenge:

- How novel is the submitted candidate's algorithm?
- How good is the computational performance of the method/algorithm?
- How does the algorithm perform in terms of precision, accuracy, etc.?

Further, to define the evaluation tree we used several inputs: (a) a questionnaire answered by the teams defining the challenges; (b) existing literature; (c) E-CEO project reports and proposal [5]; (d) several emails exchanged with the consortium domain experts setting up the challenges; (e) and also some bi-lateral teleconferences to clarify some elements to be evaluated. From these inputs, it was possible to define an evaluation tree—encompassing the three identified challenges—with three main branches at the top level, as shown in Fig. 7.3: (a) approach novelty; (b) software performance; (c) and model/algorithm performance. Each branch includes sub-criteria, represented in the 2nd, 3rd, 4th and 5th layers.

As mentioned, this figure depicts the complete tree architecture, defined for the evaluation support system of EO challenges. The first branch "novelty of approach" includes important sub-criteria to classify the novelty of the proposed application (e.g. complexity of algorithm, scalability etc.). The second branch includes the most

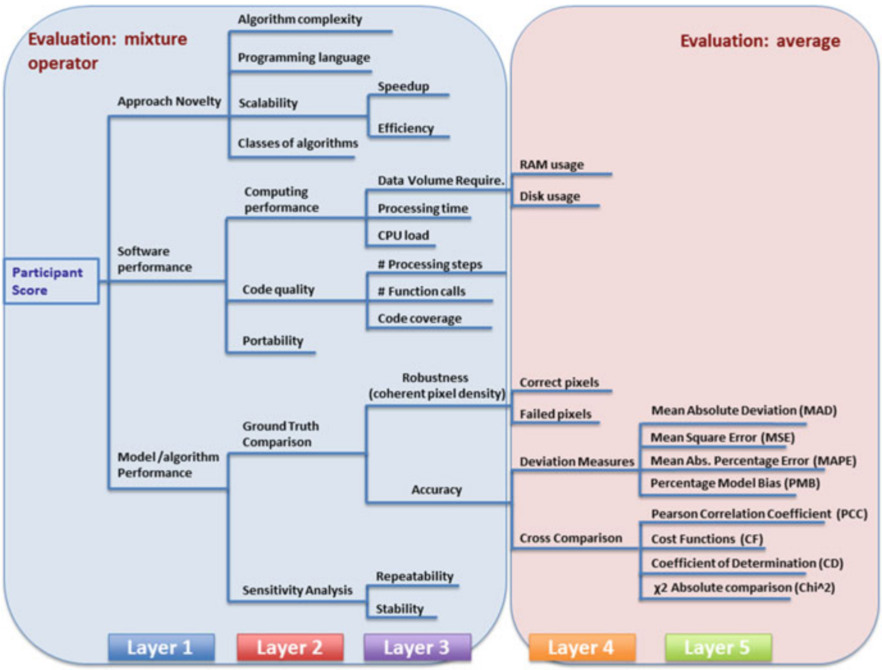


Fig. 7.3 Evaluation tree decision model

important requirements of developed software for EO manipulation: the performance of the code, its quality and its portability. The third branch includes statistical measures that are essential for measuring the model/algorithm performance in terms of ground-truth comparison. The latter is of paramount importance for EO challenges because the results achieved by the candidates must be compared with the known ground truth ("reality"), as for example: the number of correct pixels is calculated by matching the candidate provided image with the ground-truth image (provided by the initiator when a new challenge is defined).

7.3.1 *Criteria Definition*

To define each criterion, a description table was developed that included: (a) the objective of the measure; (b) unit and dimension; (c) quantification/formulation; (d) implicit logic of the criterion; (e) normalization process; (f) actor (responsible entity for the input score). A tabular form was used for the definitions to clarify and standardize the descriptions, as well as, to help the initiator define new criteria when needed for a specific challenge. To avoid lengthy details about all definitions, here we will only explain the path of one leaf node, from each of the three main branches, as follows:

(a) Software-performance/computing performance/CPU load
(b) Approach-novelty/programming language
(c) Model/algorithm-performance/ground truth/accuracy/cross-comparison/cost function

Details about the definitions for the above three leaf nodes criteria are summarized in Tables 7.1, 7.2 and 7.3. The focus is on the leaf nodes because these are the only ones that require calculations to obtain their respective scores. All criteria layers (except leaf nodes) of each branch are calculated automatically by the tool, using the proposed hierarchical synthetizing process, described in Sect. 7.5.

As can be observed in Tables 7.1, 7.2 and 7.3 this standardized format allowed formalizing different types of criteria (statistical, qualitative etc.) into an easy

Table 7.1 Software-performance/computing performance/CPU load (a)

Description of criterion CPU load				
Measure of the CPU usage during execution of algorithm				
Unit/ dimension	*Quantification*	*Quantification logic*	*Normalization (score)*	*Actor*
Percentage	Framework calculates value for each candidate software	Lower is better	S_CPU=(max − val)/(max − min)	System
Notes/comments				
There is no quantification definition because it is directly calculated by the system. The average value will be used in the quantification. *Note*: taking the maximum value would almost always result in a 100 % value—no differentiation between contestants				

Table 7.2 Approach-novelty/programming language (b)

Description of criterion programming language				
The evaluator classifies the programming languages used by challenge				
Unit/ dimension	*Quantification*	*Quantification logic*	*Normalization (score)*	*Actor*
Linguistic terms	Very novel Novel Mainstream Not novel (common)	Discrete classification	Very novel = 100 % Novel = 75 % Mainstream = 50 % Common = 5 % Ignore = 0 %	Initiator defines list Evaluator classifies
Notes/comments				
Examples: Very novel ("new" languages); Novel (e.g. Python/Ruby); Mainstream (e.g. Java/C++); Not novel (e.g. Matlab/Fortran). *Note*: Crisp values (%) are used by default and can be parameterized				

Table 7.3 Model/algorithm-perform/ground truth/accuracy/cross-comparison/cost function (c)

Description of criterion cost function				
Provides a non-dimensional value indicative of the "goodness of fit" between two sets of data; it quantifies the difference between model results and measurement data				
Unit dimension	*Quantification*	*Quantification logic*	*Normalization (score)*	*Actor*
Float	$x^2 = \frac{1}{N\sigma_D^2} \sum (M_n - D_n)^2$ M and D—model and observed variable, N—number of observations	lower is better	S_CostF = (max − val)/(max − min)	System or evaluator
Notes/comments				
Cost functions are a measure of model data mismatch and are primarily used in data assimilation, usually taking the form of the difference between model and observation, scaled by some measure of data variance. The simplest example of a cost function involves scaling RMSE (random mean square error) with the data variance				

implementation format in the tool. Further, when starting a new challenge, it is easy to verify if any new criterion or any change is required and the evaluator/initiator will define a similar table for it, which facilitates its software implementation.

Since the proposed conceptual model is based on a cross-comparison paradigm, i.e., each candidate will be compared with the best and worst candidates participating in the challenge, the framework needs to determine all criterion scores and then retrieve the max and min, to enable the proposed cross-comparison normalization. Details about this normalization process will be presented in Sect. 7.3.2.

The described sub-criteria definitions allowed their easy implementation within the framework and, therefore, automating almost all evaluation process for both qualitative and quantitative criteria. Some sub-criteria require the evaluator (or initiator) to provide the score manually, either by calculating it outside the framework or by selecting a semantic score for the respective criterion. There are only three qualitative sub-criteria, so the overload for an evaluator is rather small.

7.3.2 Normalization

Normalization is essential in most applications that require some kind of aggregation of scores or comparison of components. To perform any criteria/data aggregation or fusion, data must be both numerical and comparable; therefore, qualitative and quantitative criteria must undergo some data transformation process to ensure this comparability [17]. This is a challenging topic that could lead to very different results. In this work, two types of normalization processes were used, one for cross-comparing the results achieved in each criterion (majority of cases) and another for semantic classifications. It should be noted that in decision problems with access to grades for the worst and best candidates, the max-min normalization is usually appropriate because it allows relative classifications, i.e., the scores display how close any candidate classification is to the best classification (grade attained for a specific criterion) and how far from the worst candidate.

Max-min Normalization

The main reason for selecting the max-min normalization is that it is logically sound for comparing candidates when using as cross-reference measure such as the "best" (max) and "worst" (min) scores, obtained from all applicants, for a specific criterion. With these maximum and minimum values, a cross-comparative relative score for each candidate can be obtained. This score is normalized in the interval [0; 1], which enables data fusion in the same simple scale. For example, the sub-criteria "CPU load" should be low and therefore the logic should be "the lower the better". Conversely there are other criteria where "higher is better". Since we are using the scale [0; 1] where close to 1 is better, the two formalizations for cross-comparison normalization are:

$$\text{Lower is better: LiB} = (\text{max-score}) / (\text{max-min}) \tag{7.2}$$

$$\text{Higher is better: HiB} = (\text{score-min}) / (\text{max-min}) \tag{7.3}$$

Example Let us consider that the CPU load value of a candidate is 98 % and the best candidate obtained 50 % (min) and the worst obtained 100 % (max). Since this criterion objective is "lower is better", we have:

$$\text{LiB} = (100 - 98)/(100 - 50) = 0.04 = 4\,\%$$

Now, if the minimum score from all contestants is 90 % (probably the reality) a rating of 0.2 (20 %) will be attained for the same contestant (with 98 % of CPU load). Therefore, as can be observed, cross-comparison normalization allows relative scores related with the best and worst candidates.

Linguistic (Qualitative) Normalization

The aim of normalizing qualitative criteria is to quantify their scores in order to obtain a numerical value that can be aggregated into a single score per alternative. This type of normalization (from the realm of fuzzy set theory [17, 18]) is applied to three criteria of the evaluation tree: development software language; class of algorithms; and portability. Here, the proposed discrete and customizable semantic scale is as follows:

- Very novel/Very high = 100 %
- Novel/High = 75 %
- Mainstream = 50 %
- Common/Low = 5 %
- Ignore (Exclude) = 0 %

It should be noted that the application of a more sophisticated method could be considered, as for example, using the central point of a provided interval (e.g. in the scale 80–100 the central point is 90 %), or by asking the evaluator to use a score from the displayed intervals, e.g. 80–100 % for very high. In our case our choice was deemed sufficient.

7.3.3 Criteria Relative Importance with Weighting Functions

In multi-criteria decision problems, the question of assigning weights to express the relative importance of any specific criterion in the overall evaluation, is always a challenge [15]. Particularly, it is difficult to assign direct numerical weights to criteria because small distinctions are humanly difficult to assess. For example, for a scale of 0–1, it is rather difficult to distinguish if the weight/importance of a criterion, such as "CPU load", should be 0.4 or 0.44 or 0.42. However, humans are quite good at expressing preferences with semantic terms, because they easily grasp intervals instead of point-values; for instance, it is easy to say a criterion is Very important and another just Important. When we must transform semantic concepts (linguistic terms) into numbers, we may consider that Very important is somewhere within the interval [0.8–1] and Important within interval [0.6–0.8]. Figure 7.4 depicts the functions and the respective ranges.

Another major issue with weights, i.e., relative importance of criteria, is that weights should depend on the corresponding attribute satisfaction values [13, 15]. From a decision making perspective, when considering one criterion with a high importance, but which has a bad performance (low satisfaction value), the decisional weight of this attribute should be penalized, to render its importance less significant in the overall evaluation. Conversely, a criterion with low relative importance (weight) that has higher satisfaction values should be rewarded, thereby rendering more significant the dominance effect to that criterion. For example, (see Fig. 7.4), if a criterion is "Important" (interval $[0.6 - 0.8]$) and its score is (0.6) its importance is adjusted to around 0.72; conversely if the score is above 0.9 the weight will be around 1. The ranges can be easily customized for any specific challenge.

The simple formulation proposed for defining the functions was:

$$\text{Weight - } W_j = \text{LowerLimit} + (\text{UpperLimit - LowerLimit}) * \text{score} \tag{7.4}$$

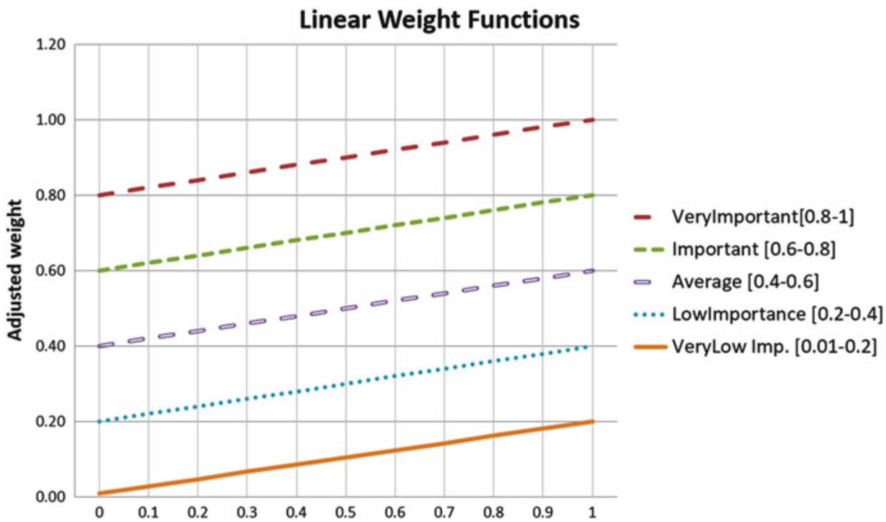


Fig. 7.4 Linear weight generating functions with ranges. X-axis criterion satisfaction values; Y-axis adjusted weights

The criterion score is considered in the weights formula to match the level of satisfaction of the criteria in the respective weight. If a score is low the corresponding weight should be as lower as possible within its relative importance, i.e., the weights penalize or reward the satisfaction of criteria (so-called mixture operators). If a criterion has score 0, its weight will be the lower limit of the assigned weight (obviously when the weighted average is calculated during criteria scores aggregation the result will be 0, but the total aggregation is divided by the sum of all weights, hence the final result will be decreased).

The above discussion expresses the philosophy chosen for expressing the relative importance of criteria (weight functions), in the hierarchical synthetizing process for layers 3, 2, 1 (shown in Fig. 7.3). This synthetizing process is based on mixture operators with weighting generating functions [13, 15] and the respective formulation is shown in Eq. (7.5). Further, Sect. 7.5.2.2 will discuss more details on how to use these weighting functions to calculate each alternative rating, i.e. how to aggregate criteria using weighting functions.

Layers 4, 5 will have equal weights because they are standard statistical measures and this implies that the aggregation of those layers will be done with a simple average.

To elicit the criteria weights (relative importance), from the evaluator, for each criterion, we use five semantic terms, which are transformed in the background into numerical classifications (within the scale [0; 1]). Section 7.4 provides a figure of the tool interface to illustrate how the relative importance is elicited (Fig. 7.5).

7.4 User Interface Design

The user interface was aimed for executing within a web browser. This provides cross-platform support, and a "run anywhere" capability.

The interface includes new visualizations modes to ensure the candidates have a user-friendly and transparent evaluation process, which also allows peer comparisons with the other candidates, both in general and also per criterion. Further, the tool interface was prepared having in mind the need for evaluation transparency—essential requirement for any open challenge.

Figure 7.5 illustrates the tool interface for eliciting the semantic relative importance for "model/algorithm performance", from the evaluator/initiator, for the three kinds of software performance. The other branches have similar interfaces to elicit all the weights for layers 1, 2 and 3 of the complete tree. It should be noted that for layers 4 and 5 there is no need to ask for weights (relative importance of criteria) because they are all statistical measures, hence, equally important.

Illustrative interfaces for displaying the candidate results can be seen in Figs. 7.7 and 7.8. Figure 7.7 shows how the candidates will see their results, in this case the scores for software performance. As illustrated, the candidates can see their own scores and compare their results with the ones obtained by the other candidates. This aspect will enable them to detect the weak and strong points of their algorithms. Figure 7.8 displays another interesting interface developed in this project—a parallel coordinates graph—with the classifications for the same branch (software performance). This visualization mode is particularly useful for cross-comparison of candidates' results because it is easy to see how a candidate behaves regarding the others, in each criterion. In Fig. 7.8 one candidate is highlighted to better illustrate this point.

Overall, the framework implementation displayed the challenge results in a user-friendly way for easy verification and peer comparison of results achieved per candidate, i.e. system alternatives. The tool was developed by the consortium leader Terradue, using a set of web development technologies, including JavaScript and

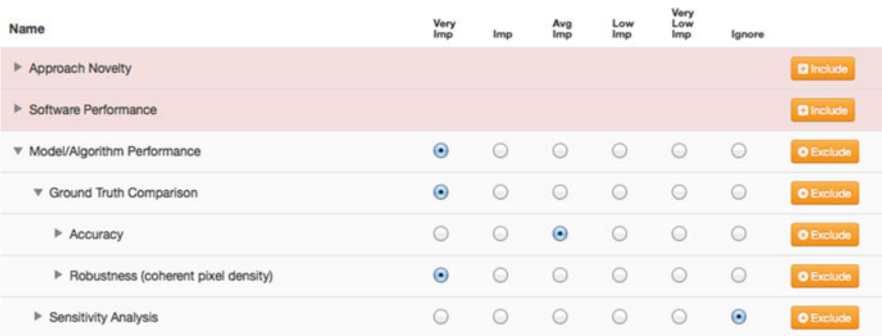


Fig. 7.5 Tool interface to elicit weights/relative importance of criteria (screenshot)

proprietary libraries. It includes many other characteristics and interface facilities, which we will not detail further, as the focus here is to discuss the novel conceptual decision model.

7.5 Evaluation Support System Process: Hierarchical Synthetizing Process for Rating

This section first describes the proposed algorithm for the hierarchical synthetizing process and then the two rating aggregation operators, selected to obtain the ratings for each layer of the three branches (Fig. 7.3), and also, for obtaining the candidates' final score.

7.5.1 Bottom-Up Hierarchical Synthetizing Process (HSP)

The HSP method includes six main steps, as schematically shown in Fig. 7.6.

1st step - When a competition is launched the initiator/evaluator chooses the criteria to be included for the competition at hand, from the evaluation tree. If any new criterion is to be added it has to undergo the "data preparation" process. If a criterion is to be deleted, the initiator will not choose it.

2nd step - The initiator/evaluator has to provide (choose) the relative importance of each criterion from the 3 first layers of the tree. This will be done using the GUI presented in Fig. 7.5.

3rd step - The evaluation process starts by calculating the internal scores for each of the evaluation-selected criteria (automatically done by the tool, e.g. Table 7.1, CPU load). Then the system stores the external rates (provided by the evaluator—e.g. Table 7.2, programming language). This internal and external rating process results in having all candidates' rates for each leaf node of the tree. From these scores, the best and worst (max and min) grades are collected to perform the normalization. The formulas for calculating the values (quantification of criteria) are illustrated in Sect. 7.3.3.

4th step - In this step the normalization process takes place. All the values from the above step are normalized with their respective method. The most common normalization uses the max and min scores to obtain the cross-reference score for each candidate. The normalization formulas proposed for each criterion are defined in Sect. 7.3.2.

5th step - Bottom-up synthetizing process for layers 4 and 5. First, scores all sub-criterion of layer 5, are aggregated and then the same process is done for each criterion of layer 4, using the calculated values of layer 5 (when they exist). The respective non-weighted normalized score values are nevertheless applied in the case of a leaf node by using a simple average.

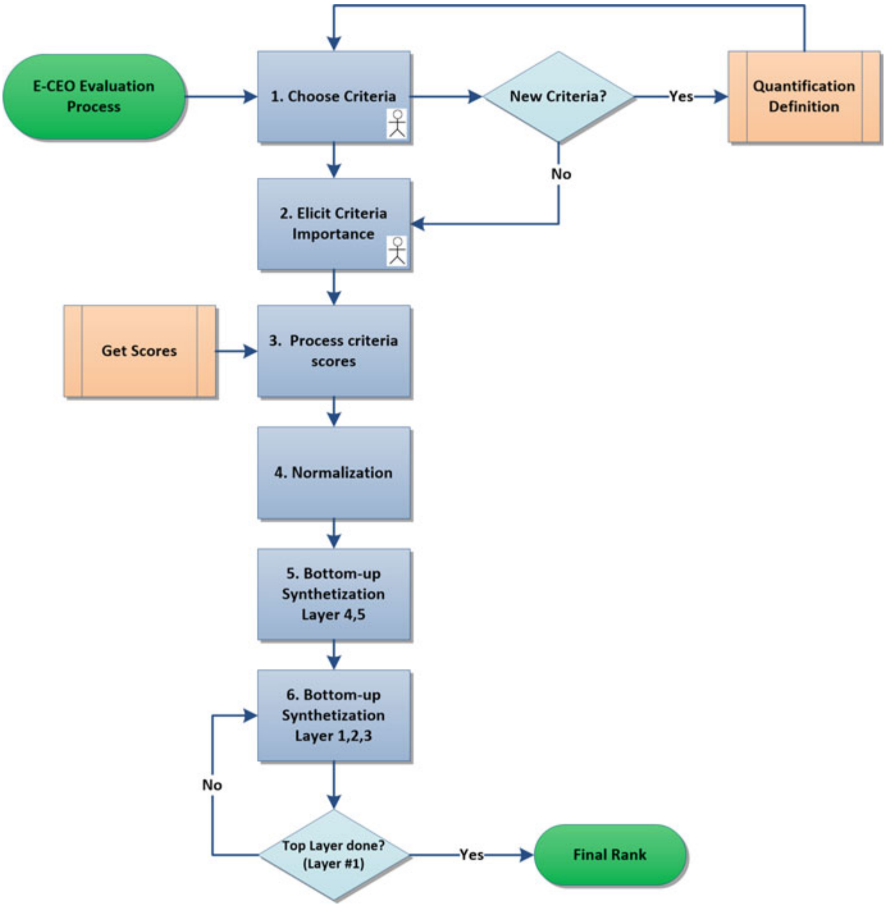


Fig. 7.6 Steps for evaluation decision model

6th step - Bottom-up synthetizing process for layers 3, 2, 1. With the results obtained from the aggregation of the respective layers below, 4 and 5, and their respective weights obtained by matching the satisfaction scores in the weight functions (Fig. 7.4), all corresponding sub-criteria are aggregated per layer until reaching the final score of the candidate in layer 1. This process is detailed in Sect. 7.5.2.

In summary, to obtain the final rating for each candidate and also its respective score on each criterion the six steps of the proposed bottom-up hierarchical synthetizing process are used. The process starts by aggregating the leaf values (Fig. 7.3) of layer 5 for each sub-criteria in layer 4, then the scores obtained for components of layer 4 (for each sub-criteria in layer 3) are aggregated and so on, until the final rating value is obtained for each candidate from aggregating scores for the three top level criteria in layer 1. With this method, the framework is able to provide feedback for the candidates, regarding their own classifications and also their ranking position towards the other candidates, plus obtaining a final score that represents the aggregated value of all criteria classifications.

7.5.2 Rating Process

There will be two different rating (aggregation) methods. One is a simple average for layers 4 and 5 because it corresponds to aggregating statistical measures with the same priorities (relative importance). The second aggregation process uses weighting functions to allow rewarding or penalizing the score obtained depending on its level [15].

7.5.2.1 Rating Layers 4 and 5 (Step 5)

Layer 4 and 5 will be aggregated with a simple average because: (a) layer 5 only includes statistical measures hence they are all considered equally important; (b) in layer 4, the measures used (e.g. correct pixels versus failed pixels, deviation measures versus cross-comparison) are also considered equally important because they represent a "precision" measure. For instance, it is equally important to evaluate the number of correct pixels and the number of incorrect pixels and perform a cross-comparison of results with the other candidates. Formally, the average rating for layers 4 and 5 is simply:

$$L_j = \frac{\sum_1^n c_{ij}}{n} \tag{7.5}$$

where n is the number of chosen criteria for specific challenge (defined by the initiator) and c_{ij} is the score of criterion i of layer j.

After obtaining the value for L_j in layer 5 these will be aggregated, again with the averaging aggregation operator [Eq. (7.5)], to obtain the rating for layer 4. Next, the three remaining layers to be aggregated will use mixture operators with weighting functions [15] (i.e. weighted average with weighting functions, as described in the next section).

In summary, this proposed hierarchical rating process is a bottom-up approach that enables a simple combination of criteria and respective sub-criteria evaluations, from any decision evaluation tree, to obtain a final score for each alternative (in our case challenge candidates).

7.5.2.2 Rating Layers 1, 2 and 3 (Step 6)

For layers 1, 2, 3 the scoring formula is more sophisticated to enable further distinction of criteria and sub-criteria, especially considering the EO competition contexts could be quite varied. Depending on the challenge, the initiators might place more importance (weight) on certain criteria than others; hence the satisfaction (classification obtained by the contestant on a particular criterion) should be emphasized for all important criteria.

As mentioned before, the criteria for each layer considered will be aggregated using mixture operators with weighting functions [13, 15] to discriminate (penalize) the criteria with low satisfaction values from those that should be rewarded. It should be noted that mixture operators with weighting functions are an extension of the classical weighted average, but where the weights are dependent on the criteria satisfaction value, instead of a predefined value. More information about the mixture operators with weighting functions (i.e. averaging aggregation with weighting functions) can be seen in the following references: [13, 15].

The mathematical formulation for aggregating layers 3 and 2 is:

$$L_j = \frac{\sum_{i=1}^{n} w_j(c)c_{ij}}{\sum_{j=1}^{k} w_j(c)} \text{ only for } c_{ij} \neq n/a \tag{7.6}$$

where:

- Rating L_j is the aggregation value for layer j;
- c_{ij} is the score of criterion i of layer j;
- $c_{ij} \neq n/a$ ensures that only any discarded criterion will not be considered in the calculation;
- $w_j(c) = f_j(c_{ij})$, where $j = 1, \cdots, n$ and $w_j(c)$ are the weights obtained from the linear weighting functions for the respective criterion j [Eq. (7.5)]. Only $w_j \neq n/a$ will be applied in the calculation.

Note: In layers 3, 4, 5 the non-chosen criteria for the specific challenge (defined by initiator) will not enter in the rating [Eq. (7.6)]. This is to prevent dividing by weights without corresponding scores (i.e. criteria without score).

7.6 System User Experience (Experimental Cases)

We first extracted a branch sector from the evaluation tree (Fig. 7.3) to demonstrate, in a step-by-step manner (flowchart), how the method works (Fig. 7.6) and to clarify the proposed approach. Second, we describe part of an experimental case, corresponding to the same branch as in the first example, which was tested directly in the tool platform. Both illustrative examples were prepared by a domain expert of the project [5, 11] and acted as a validation case study for the implemented tool. The second case illustrates how the candidates are ranked; how they perceive their results and how they can compare themselves with their competitors (peer comparison).

A significant amount of tests and validations were performed during the project, to assess the robustness and adaptability of the model and respective implemented tool, however, in this work the focus is on clarifying how the tool works and how it depicts the results for the candidates.

7.6.1 Example for Demonstrating Step-by-Step Method

The illustrative example was prepared using the computing performance sub-branch of the software performance branch, because it includes common cross-cutting criteria to any software performance evaluation. Further, this example demonstrates the two types of aggregation in the bottom-up synthetizing process.

Let us consider that one contestant obtained the following scores after running his/her submitted software (Table 7.4):

In this case since all sub-criteria objective (logic) is to have the lower value possible the "ideal" candidate is the one with the minimum values: 90 % CPU load; 1200 s for processing time; and 128 MB for both memory (RAM) and disk usage.

1st step: is considered done because the criteria were chosen a priori.
2nd step: elicit criteria importance (weights).

Here the initiator is asked to provide the semantic weights for all criteria in layers 1, 2 and 3, using an interface similar to the one implemented in the tool (illustrated in Fig. 7.5), as shown in the next table. In this example there are only 3 criteria to be evaluated and the elicitation of weights were as follow (Table 7.5):

3rd step: normalize values.

Table 7.4 Small example for computing performance with one candidate and the worst and best scores of all other candidates

Computing performance				
		Processing	Data volume	
	CPU load	time (s)	RAM usage (MB)	Disk usage (MB)
Contestant classification	95 %	1700	256	512
Best contestant (Max*)	90 %	1200	128	128
Worst contestant (Min*)	99 %	2000	1024	1024

Note: the Min and Max are respectively the lower and higher values obtained from all competitors

Table 7.5 Inputting the semantic weights for criteria

⋯ Computing performance	Very imp	Imp	Avg. imp	Low imp	Very low imp	Ignore
CPU load	○	•	○	○	○	○
Processing time	•	○	○	○	○	○
Data volume requirements	○	○	○	•	○	○

The proposed normalization allows each contestant to have a relative order of its performance in each criterion, i.e., each sub-criterion classification is relative to the ones obtained by the “best” and “worst” candidates that applied to the challenge. For example, for CPU load the candidate knows that the best contestant got 90 % (classification 1) and the worst one got 99 % (classification 0). With a classification of 0.444 for CPU load, the contestant knows his classification is not very good (below average) when compared with the “best” candidate (90 %). Similarly, we can provide the same rational for each of the other sub-criteria classification (Table 7.6).

4th step (Layer 4): determine challenge rates.

For this step, the relative weights are needed and then the rating for layer 4 (aggregation of sub-criteria in layer 4, with respect to complete tree in Fig. 7.3) are determined. In layer 4 the weights are assigned a priori (equally important) hence the rating for data volume is done with a simple average, as follows:

$$\text{Data-volume} = (0.857 + 0.571)/2 = 0.714$$

Hence, we now have the following ratings for layer 3 (Table 7.7):

5th step (Layer 3): Bottom-up synthetizing.

To do the bottom-up synthetizing for the criteria “computing performance” (aggregation from layer 3), first the relative weights for each criterion need to be determined. For example, since CPU load is “Important” (see weight function in Fig. 7.4) the weight 0.69 is obtained for the respective satisfaction rating 0.444. In this fashion, the classification obtained for any criterion is either penalized or rewarded, i.e. if the criterion satisfaction is high the weight will be higher (within its respective range, obviously). The weights obtained for the three sub-criteria are (Table 7.8):

Now, to obtain the final rating for criterion “Computing performance” layer 3 sub-criteria (CPU load, Processing time, Data volume requirements) are synthetized with the mixture operator [Eq. (7.6)], as follows:

$$\begin{aligned}\text{Computing performance} &= (0.444 * 0.69 + 0.375 * 0.88 + 0.714 * 0.34)/\\ &\qquad (0.69 + 0.475 + 0.34) = 0.46\end{aligned}$$

Table 7.6 Criteria values after normalization

	CPU load	Process. time (s)	Data volume	
			RAM usage (MB)	Disk usage (MB)
Contestant classification	0.444	0.375	0.857	0.571

Note: normalization was done with the formula presented in Table 7.1

Table 7.7 Data volume rating calculated

	CPU load	Process. time (s)	Data volume
Contestant classification	0.444	0.375	0.714

Table 7.8 Results after weighting

	Important	Very important	Low importance
	CPU load	Process. time (s)	Data volume
Satisfaction weighted	0.69	0.88	0.34

Note: the same procedure would take place in the complete model until the layer 1 is reached (i.e. by synthetizing layer 2 the results for layer 1 are obtained; the final score).

When the initiator/evaluator does not want to consider a criterion for a specific challenge, the evaluation model works without any problem. For example, let us consider the evaluator does not choose sub-criterion "processing time". Then the final calculation for the synthesizing process will simply be:

$$\text{Computing performance} = (0.444 * 0.69 + 0.714 * 0.34)/(0.69 + 0.34) = 0.534$$

7.6.2 *Illustrative Case for Demonstrating Peer Comparison of Results*

Pixalytics personnel (one of the project Consortium companies, henceforth denoted PIX) are the EO experts who prepared a test bench challenge using the conceptual model defined by UNINOVA. The prepared challenge included 15 contestants, with the respective input values for each candidate's criteria, plus the weights / relative importance for all criteria as well as the scores for qualitative criteria. Finally, PIX also determined the scores, ratings and final ranking of all contestants to be validated in the developed tool. It is important to note that some criteria (Fig. 7.3) were not selected by PIX, as for instance "sensitivity analysis" of the model performance branch. Figure 7.7 shows the results obtained for the illustrative branch with all selected sub-criteria and the global score (first left result column). For simplicity, the figure only displays the results for the first 8 ranked candidates.

Observing the figure, specifically the sub-criterion "processing time", the score is identical to the one obtained in the small example, i.e. 0.375. This value is obtained from the input values in Table 7.4, using the normalization formula [Eq. (7.2)] such as: $0.375 = (2000 - 1700)/(2000 - 1200)$. In the case study, the input values were identical but the weights were different. All other criterion values and weights were different because the domain expert provided them. Further, as can be seen in Fig. 7.7, this case study included all criteria from the software performance branch.

Another interesting characteristic of the tool is the user-friendly interface, which enables peer comparisons in a simple and straightforward manner. Figure 7.8 depicts the parallel coordinates graph, developed to allow peer comparison. Again, here only the software performance branch results for illustrating this tool facility are shown. The results pertaining to the 8th ranked candidate, ajf@uninova.pt, are highlighted.

	username	globalScore	Software Performance	Computing Performance	CPU Load	Processing Time	Data volume requirements	RAM usage	Disk usage	Code Quality	Number Processing steps	Number Function calls	Cod Cov
	cdangelis	0.69968	0.5626051	0.4026095	0	0.875	0.2657143	0	0.5714286	0.6666667	1	0	0.66
	fbrito	0.6525	0.8692708	1	1	1	1	1	1	0.8333333	1	1	0.33
	emathot	0.58452	0.8623529	1	1	1	1	1	1	0.8	1	1	0
	pgoncalves	0.58357	0.5487133	0.3862691	0	0.5	0.8571429	1	0.7142857	0.9019608	1	1	0.66
	fcerasuolo	0.56812	0.650753	0.4749963	0.3333333	0.375	0.8571429	1	0.7142857	0.9019608	1	1	0.66
	webtest-participant-2	0.56629	0.8498893	0.950758	1	1	0.8035714	0.75	0.8571429	0.8333333	1	1	0.33
	fdandria	0.54544	0.7777008	0.9284619	1	0.875	0.8571429	1	0.7142857	0.8333333	1	1	0.33
	ajf@uninova.pt	0.54006	0.5644448	0.407014	0.3333333	0.375	0.6071429	0.5	0.7142857	0.6666667	1	0	0.66

Fig. 7.7 Tool interface of results for software branch Software Performance (screenshot)

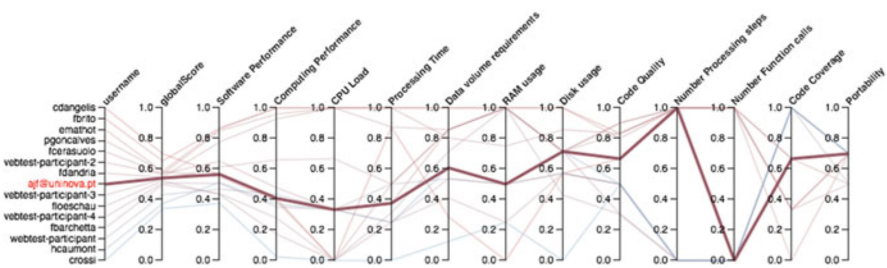


Fig. 7.8 Tool interface of peer comparison for Software Performance branch (screenshot)

7.7 Conclusions

In this chapter we discussed a novel model-driven evaluation support system for open challenges within EO domains. The proposed system allows competition initiators/evaluators to define a specific challenge (competition) and then easily adapt the hierarchical structure (tree) to add or delete more criteria or sub-criteria, if needed. After, it is quite straightforward for the evaluators to select the qualitative of quantitative formulations—provided by the tool—that best suits the criteria evaluation. After the candidates submit their proposals, the ESS calculates the ratings for each criterion and determine their final score. All candidates can also visualize their ratings and perform peer comparison with results from the other candidates. Specifically, in this book chapter we discussed the evaluation decision model for EO open challenges, which combines a hierarchical bottom-up multi-criteria logical structuring with a fuzzy fusion process to achieve the ranking of candidates. Furthermore, the evaluation decision model allows transparent peer comparisons of candidates and their comparative ranking with other candidates, in the overall challenge.

In summary the main characteristics/advantages of the evaluation support system are:

(a) It provides ratings per sub-criterion and then by synthetizing, calculates the scores for each layer of the evaluation tree, in a bottom-up process.
(b) The evaluation procedure is straightforward and simple to explain to contestants.
(c) Provides a cross-comparison (relative rating) of each contestant regarding the "best and worst candidates"—for each branch and respective sub-criteria in tree—due to the normalization process used. This allows the contestants to see their comparative ratings and assess their performance regarding the "best" and "worst" results of the current challenge.
(d) The aggregation process of layers 4 and 5 is straightforward (average) and its rational is obvious.
(e) The aggregation process of layers 1, 2, 3 is a bit more complex, but considers penalizing or rewarding the satisfaction of each criterion in relation to its relative importance (weight).
(f) Even when one or more criterion (or even a complete tree branch) is not included for evaluation (i.e. it was deleted by the initiator/evaluator), the evaluation process is robust and runs smoothly.
(g) If a new criterion needs to be added to the evaluation tree, the initiator will define it, as shown in Sect. 7.3 (normalization, quantification and weighting functions, if needed).

The ESS tool itself provides a user friendly interface allowing the initiator to define the challenge criteria in a straightforward manner. It provides a helpful interaction method for the candidates and for the evaluators an invaluable decision tool in the evaluation process of the EO challenge.

Concluding, the discussed decision model of the evaluation support system is general (i.e. can be used for many types of competitions), as well as adaptable and robust (i.e. allows adding or deleting criteria and the score calculation uses a robust calculation process), for any kind of challenge within the EO domain.

Acknowledgements This work was partially funded by the European Space Agency (Contract No. 4000107471/13/I-BG). This work was partially funded by FCT Strategic Program UID/EEA/00066/203 of UNINOVA, CTS. The authors would like to thank all the support of the ESA technical officers Jordi Farres and Michele Iapaolo, and all the project consortium members.

References

1. Calvo, T., Mayor, G., Mesiar, R.: Aggregation Operators: New Trends and Applications, vol. 97. Physica, Heidelberg (2012)
2. Chang, Y.H., Yeh, C.H., Chang, Y.W.: A new method selection approach for fuzzy group multicriteria decision making. Appl. Soft Comput. **13**(4), 2179–2187 (2013)
3. Danielson, M., Ekenberg, L.: Rank ordering methods for multi-criteria decisions. In: Group Decision and Negotiation. A Process-Oriented View, pp. 128–135. Springer, Berlin (2014)

4. de Almeida, A.T.: Multicriteria decision model for outsourcing contracts selection based on utility function and electre method. Comput. Oper. Res. **34**(12), 3569–3574 (2007)
5. E-CEO: E-CEO Project financed by ESA, coordinator Terradue Ltd (2013). http://wiki.services.eoportal.org/tiki-index.php?page=ECEO and http://www.ca3-uninova.org/project_eceo
6. E-CEO: Web Portal. https://challenges.esa.int/ (2016) [Online]. Accessed 20 May 2016
7. Espinilla, M., Lu, J., Ma, J., Martínez, L.: An extended version of the fuzzy multicriteria group decision-making method in evaluation processes. In: Advances on Computational Intelligence, pp. 191–200. Springer, Berlin (2012)
8. Hobbs, B.F., Chankong, V., Hamadeh, W.: Does choice of multicriteria method matter? An experiment in water resources planning. Water Resour. Res. **28**(7), 1767–1779 (1992)
9. Kou, G., Peng, Y., Lu, C.: MCDM approach to evaluating bank loan default models. Technol. Econ. Dev. Econ. **20**(2), 292–311 (2014)
10. Kumar, D.N.: Multicriterion Analysis in Engineering and Management. PHI Learning Pvt. Ltd., New Delhi (2010)
11. Lavender, S., Brito, F., Aas, C., Casu, F., Ribeiro, R., Farres, J.: E-Collaboration for Earth Observation (E-CEO) with the example of contest# 3 that focuses on the atmospheric correction of ocean colour data. In: EGU General Assembly Conference Abstracts, vol. 16, p. 4503 (2014)
12. Linkov, I., Satterstrom, F., Kiker, G., Batchelor, C., Bridges, T., Ferguson, E.: From comparative risk assessment to multi-criteria decision analysis and adaptive management: recent developments and applications. Environ. Int. **32**(8), 1072–1093 (2006)
13. Pereira, R.A.M., Ribeiro, R.A.: Aggregation with generalized mixture operators using weighting functions. Fuzzy Sets Syst. **137**(1), 43–58 (2003)
14. Ribeiro, R.A.: Fuzzy multiple attribute decision making: a review and new preference elicitation techniques. Fuzzy Sets Syst. **78**(2), 155–181 (1996)
15. Ribeiro, R.A., Pereira, R.A.M.: Generalized mixture operators using weighting functions: a comparative study with WA and OWA. Eur. J. Oper. Res. **145**(2), 329–342 (2003)
16. Ribeiro, R.A., Moreira, A.M., Van den Broek, P., Pimentel, A.: Hybrid assessment method for software engineering decisions. Decis. Support. Syst. **51**(1), 208–219 (2011)
17. Ribeiro, R.A., Falcão, A., Mora, A., Fonseca, J.M.: FIF: a fuzzy information fusion algorithm based on multi-criteria decision making. Knowl.-Based Syst. **58**, 23–32 (2014)
18. Ross, T.J.: Fuzzy Logic with Engineering Applications. Wiley, New York (2009)
19. Triantaphyllou, E.: Multi-Criteria Decision Making Methods: A Comparative Study, vol. 44. Springer Science & Business Media, Berlin (2013)
20. Turban, E., Aronson, J., Liang, T.P.: Decision Support Systems and Intelligent Systems, 7th edn. Pearson/Prentice Hall, Upper Saddle River, NJ (2005)
21. Tzeng, G.H., Huang, J.J.: Multiple Attribute Decision Making: Methods and Applications. CRC, Boca Raton, FL (2011)
22. Yoon, K.P., Hwang, C.L.: Multiple Attribute Decision Making: An Introduction, vol. 104. Sage, Thousand Oaks, CA (1995)

Chapter 8
An Optimization Based Decision Support System for Strategic Planning in Process Industries: The Case of a Pharmaceutical Company

Narain Gupta and Goutam Dutta

Abstract In this research, we present the design, development, implementation and testing of a model-based DSS for strategic planning in process industries. This DSS was first developed as a single period model, later it was extended to multiple period planning and then multiple scenario planning. We demonstrate how a complex process industry like the pharmaceutical industry can be modelled using stochastic linear programming (SLP). We describe how a generic, user friendly, menu driven model-based DSS can be designed and developed. We also demonstrate that such systems can be used by managers with no or little knowledge of OR/MS. The DSS is tested using real data from a pharmaceutical company. We demonstrate the impact of modeling uncertainty using SLP. We conduct a set of optimization experiments in order to demonstrate the impact of stochastic optimization. The impact of the optimization and modeling uncertainty is measured in terms of a Value of Stochastic Solution (VSS). The successful testing of the DSS with real data from a pharmaceutical company demonstrates a potential bottom line impact of 5.36 %, equivalent to USD 1.26 million. We discuss the characteristics of a good model-based DSS. We present the key features of our DSS in this context. We also discuss the reasons behind the failures of several DSSs in practice. We conclude the chapter by sharing the lessons that we learnt in our journey of development and use of the proposed DSS.

N. Gupta (✉)
Management Development Institute, Gurgaon, 122001 Haryana, India
e-mail: Narain.gupta@mdi.ac.in

G. Dutta
Indian Institute of Management, Ahmedabad, Ahmedabad 380015, India
e-mail: goutam@iimahd.ernet.in

© Springer International Publishing Switzerland 2016

J. Papathanasiou et al. (eds.), *Real-World Decision Support Systems*,
Integrated Series in Information Systems 37,
DOI 10.1007/978-3-319-43916-7_8

8.1 Introduction

The healthcare and pharmaceutical sectors are rapidly growing businesses in India. The time span to conclude a new drug in the market can be as high as 15–20 years, with stiff competition among pharmaceutical companies. Companies regularly look for avenues to come up with newer molecules in the market. With shrinking margins in a limited marketplace, companies have started spending heavily on cost efficiencies through strategic sourcing, efficient supply chains and optimal product mix approaches.

The manufacturing of finished products requires the processing of raw materials through a complex network of production facilities. The capacity of the facilities available and needed for each finished product also varies substantially. By varying the product mix, the net (contribution to) profit can be maximized.

The managers of the pharmaceutical companies plan their production based on historical data and their judgement. May researchers proposed mathematical models to solve the problem that the pharmaceutical companies are interested. The literature reports the design and development of a series of DSS for determining the optimal product mix in different industries. In the last two decades, there has been a growing need to apply optimization-based tools to pharmaceutical manufacturing. The implementation of such optimization models and DSS has been difficult due to several factors including the resistance to organizational change and the lack of availability of a user-friendly system. Also, managers do not seem to trust the decision support provided by the DSS.

This research is motivated by earlier reported studies in various process industries including steel industry optimization by Fourer [10], multi-period optimization by Dutta [5], and Dutta and Fourer [4], pharmaceutical industry production optimization by Dutta et al. [6], a multi-period optimization in various process industries by Dutta et al. [7], and the research on the design and development of a multi-scenario DSS by Gupta et al. [14].

In this study, we design and develop a multi-period, multi-scenario DSS for production planning in the pharmaceutical industry. We discuss the user-friendliness of the DSS, which would help managers to use such planning tools in practice. The DSS is developed and implemented in an RDBMS platform called 4th Dimension.

In this study, we describe the difficulties that arose during the design and development of the DSS. We also look at the issues related to the implementation of the DSS. We conclude the study with the lessons that future managers and researchers may learn from the development of this DSS. A set of experiments have been designed and conducted using data from a pharmaceutical company. In this study, we attempt to address the questions given below in the context of the pharmaceutical industry:

1. How can we model the uncertainty in the demand of the finished product using the SLP?
2. What is the impact of modeling the uncertainty in the demand of finished product using SLP?

3. What is the value of expected value of perfect information (EVPI) compared to the value of the stochastic solution (VSS)?
4. What are the key characteristics of a good model-based DSS?
5. What are the design and implementation challenges from the modeller's and the end-user's perspective?

The structure of this chapter is as follows. The motivation of this research is stated in Sect. 8.2. Section 8.2 also reviews the relevant literature in the field of model-based DSS applications in pharmaceutical companies. For the readers new to the field of stochastic programming, we provide a brief introduction and an illustrative example in Sect. 8.3. The design and development of the DSS and the modeling of pharmaceutical production processes is discussed in detail in Sect. 8.4. Section 8.5 describes the application of the mathematical model and the DSS in a pharmaceutical company. The design of optimization experiments is also detailed in Sect. 8.5. The stochastic optimization results are presented and discussed in Sect. 8.6. Section 8.7 presents the overall user experience. The lessons learnt from our journey of design and development of this model-based DSS, key characteristics of a good model-based DSS, and the key challenges in a model-based DSS are also elaborated in Sect. 8.7. We conclude the chapter in Sect. 8.8.

8.2 Literature Review and Motivation

In this chapter, we present the relevant reported literature for similar DSS and real-world applications. We describe a few important indicative studies and the novelty of the proposed DSS.

8.2.1 Literature Review on Real-World Applications of a DSS

Several applications of mathematical programming in process industries in general, and the pharmaceutical industry in particular, have been reported in literature. Some examples include a genetic-based facility layout method by Hamamoto et al. [15], a queuing network modeling and lead time compression of pharmaceutical drug development by Viswanadham and Narahari [25], facility location and distribution planning by Gupta et al. [13], an integrated production and transportation scheduling model for two geographically separated plants by to Miller and Matta [18].

The pharmaceutical industry is highly capital and R&D intensive. According to Viswanadham and Narahari [25], the success of any pharmaceutical company depends on its ability to successfully develop and market new drugs, faster than the competition. The company can enjoy the benefits of patented drugs in a monopolistic market during the patent validity period. The DSS proposed by Gupta et al. [13] was implemented at Pfizer/Warner—Lambart, which is in the pharmaceutical and health care sector. The DSS is capable of operational planning, medium term and strategic planning for product distribution and facility location.

Next, we review the recent literature on production planning in a deterministic and uncertain environment using a DSS. Venkatadri et al. [24] address the usefulness of a DSS which helps the firms commit orders to a due date. The DSS demonstrates the update of the production, warehousing and distribution plans based on price and due date negotiations. This DSS was based on information sharing in advance and in a deterministic environment. Galasso et al. [12] proposed a framework to simulate a rolling production planning process under a flexible demand situation. This is an application of modeling uncertainty without a DSS. Xue et al. [27] presented a multi-objective optimization algorithm for product development and manufacturing planning. The model was presented in the context of a printed circuit board assembly industry in a deterministic situation. Fazlollahtabar et al. [8] proposed a service-oriented DSS framework. The framework helps to analyze an ERP under changing market situations in order to be agile and flexible.

Miller et al. [19] constructed a DSS that utilizes databases, programming languages and spreadsheets from the firm's ERP. The DSS was designed to help Pfizer's supply chain network planning to guide the daily shipments and improve its inventory control. Mansouri et al. [17] review the literature of two decades on multi-objective optimization for build-to-order supply chain management. The review highlighted the gap in the development of the model, optimization techniques and DSS.

Scott et al. [22] proposed an integrated method for modeling uncertainty in demand and supply using the chance constraint programming and analytical hierarchical process. The chance constraint programming is one of the SLP approaches. They developed a DSS on this integrated model. Palanisamy et al. [20] presented a DSS in an assembly line environment for line balancing. The DSS was reported to have optimized the production cost and the number of workstations. The model was built under deterministic constraints. Mohammad et al. [21] demonstrated the use of a simulation based DSS for container space utilization in a logistics company. It was a study in a deterministic environment for improved collaboration among supply chain partners.. There are very few reported applications which integrate on modeling the demand under uncertainty, application of SLP, and a DSS. The studies in literature propose either models in a deterministic environment or simulation based DSS.

8.2.2 Motivation for the Development of the Proposed DSS

A series of publications by Dutta et al. [5], Fourer [10], Dutta and Fourer [4], Dutta et al. [6], Dutta et al. [7] and Gupta et al. [14] demonstrate the design and development of an optimization-based DSS and its application in different process industries. The applications range from a steel company in North America by Dutta [5], Dutta and Fourer [4], to a pharmaceutical company in Western India by Dutta et al. [6], and further to an aluminum company in Eastern India by Dutta et al. [7]. The DSS described in Dutta and Fourer [4] demonstrated a potential increase of

16–17 % in the bottom line of the company. Similarly, the testing and application of an LP-based DSS with multi-period real data of an integrated aluminum company and a pharmaceutical company demonstrated a potential impact of 6.72 % and 12.46 %, respectively. This high impact potential motivated us to study the database construction principles, and application of stochastic optimization-based DSS for the process industry in general, and the pharmaceutical industry in particular.

Dutta et al. [6] presented a study of product mix optimization at three manufacturing units for liquid syrups, tablets and injections in an Indian pharmaceutical company. Readers may refer to Wallach and Loisel [26] and Varlan and Paillier [23] for other reported applications in pharmaceutical companies.

Fourer [10] proposed a single-period model with three fundamental elements, namely *Materials*, *Facilities* and *Activities*. This work was extended by Dutta and Fourer [4] for multiple-period planning with two additional elements, namely *Storage Areas* and *Times*. In later extensions, the database design principles and SLP-based DSS were introduced and discussed by Gupta et al. [14]. The complexity involved in the design of a hierarchical database structure in comparison to a completely relational one for the development of a multi-period optimization based DSS was discussed in these extensions.

We demonstrate that a generic, user-friendly SLP-based DSS can be used for planning in a probabilistic demand situation. We design a set of experiments based on the variability in the demand of finished pharmaceutical products. In this study, the proposed DSS has been tested with real data from a pharmaceutical company. However, since it is generic, it can be used to model problems in many other similar process industries. This research primarily focuses on modeling uncertainty in the demand of finished goods. However the multi-scenario, multi-period DSS is capable of simultaneously modeling uncertainty in a number of parameters of the model including the demand for finished goods, the cost of the purchase of raw material, the sell price of finished goods, the supply of raw materials, etc. This is probably one of the early attempts when a generic, user-friendly, multiple-period SLP-based DSS has been designed, developed and tested with real- world data from a pharmaceutical company.

Two sets of audiences have been targeted while articulating this study. First, scholars who plan to study modeling under uncertainty and its integration with dialog management, data management, and model management. Second, researchers and practitioners who plan to study and implement a model-based DSS in complex industries like the pharmaceutical manufacturing industry.

8.3 Stochastic Linear Programming (SLP): An Illustrative Model

There is an extensive literature available on production planning, optimization, DSS, databases, etc., where the emphasis has been on the model, data and solver independence. However, the literature on modeling uncertainty and the user-friendliness of a model-based DSS in complex industry operations is limited. Also, there are limited reported applications of two-stage SLP with recourse to model the process industry including the uncertainty on the model parameters. The term recourse is defined by Fragniere and Gondzio [11] as the adaptability of decision variables to different outcomes of the random parameters at each time period. In an SLP with recourse, the response of the randomness of the model is corrected as a part of the model. We introduce SLP using the first deterministic equivalent linear program (DELP) formulated by Dantzig [3] for the audience new to this field of knowledge:

$$\begin{array}{llllll} \min & c_1X_1 & + p_1c_2X_2^{(1)} & + p_2c_2X_2^{(2)} & + p_3c_2X_2^{(3)} & \\ s.t. & A_{11}X_1 & & & & = b_1 \\ & A_{21}X_1 + & A_{22}X_2^{(1)} & & & = b_2^{(1)} \\ & A_{31}X_1 + & & + A_{33}X_2^{(2)} & & = b_2^{(2)} \\ & A_{41}X_1 + & & & + A_{44}X_2^{(3)} & = b_2^{(3)} \end{array}$$

where X_1 and X_2 are nonnegative decision vectors for all scenarios.

An SLP with three scenarios is presented above. Here, p_1, p_2, p_3 are the probabilities of the scenario occurring. X_1 is the first stage decision vector. X_2 is the second stage decision vector. The superscript of the second stage decision vector denotes the decisions in each of the scenarios. The matrix of technological coefficients is denoted by A and is assumed to be deterministic. The cost vectors of the first stage and second stage are c_1 and c_2. The random vector that varies in different scenarios is denoted as b_2.

8.4 Decision Support System

8.4.1 Modeling the Pharmaceutical Industry's Production Operations

The issues in modeling a problem for the pharmaceutical industry are the same as those faced in process industry modeling. We describe the characteristics of a typical process industry and the model used in this DSS as follows: Process industries are characterized as having fewer raw materials (such as chemicals) and a large number of finished products (such as pharmaceutical drugs). The processes are continuous in nature, so there is no work in progress material. The raw materials are processed through a network of machines, wherein the intermediates usually can neither be bought nor be sold. All materials can be inventoried and used later. In our model, the buying, the selling or the inventorying of any product can be controlled by setting bounds on their corresponding decision variables.

8.4.1.1 Fundamental Elements of Process Industry Production System

The typical fundamental elements of a process industry production system are *Materials*, *Facilities*, *Activities*, *Times*, *Storage Areas* and *Scenarios*:

- **Materials** are the physical items that are used in any of the production stages. These include raw materials, intermediates and finished products.
- **Facilities** are the collection of machines that produce one of multiple materials from other machines.
- **Activities** are the productive transformation of the materials. The production of an intermediate or a finished product at a facility is defined as one activity. Each facility houses one or more activities, which use and produce materials in certain proportions. We assume the production system to be continuously linear and hence we use linear models.
- **Times** are the periods of the planning horizon, represented by discrete numbers $(1, 2, 3, \cdots)$. The duration of planning periods can vary from weeks to years depending upon the planning horizon.
- **Storage Areas** are the places where raw materials, intermediates and finished goods can be stored.
- **Scenarios** are the possible outcomes of a hypothesized event. Scenarios are represented by a name attached with a probability of occurrence. For example, the economy at the end of the year may remain high or low with 0.6 and 0.4 probability of occurrence.

8.4.1.2 Model Assumptions

In order to develop optimal and meaningful results, the model is developed under a set of assumptions. These assumptions are based on the characteristics of process industries and the generic nature of the model. As already discussed in Sect. 8.2, this model is an extension of the single-period model presented in [4, 10]. Interested readers may refer to the aforementioned papers for a detailed understanding of the model and assumptions. In this chapter, we describe the assumptions briefly as follows:

1. Each material including raw materials, intermediates and finished products, is allowed to be purchased, stored or sold at each stage of the production. This makes the model generic, while we control the sale of raw materials, storage of intermediates, and purchase of finished products by their upper bounds.
2. The model parameters such as purchase cost, sale price, inventory carrying cost, facility operating cost, activity operating costs, facility capacities, storage capacities, etc., may change over different production stages and different time horizons.
3. At any time, one or more materials are used as input and output in a facility. Generally, more than one material is used to produce one product. The relative proportion of various inputs and outputs (generally called technological coeffi-

cients) in an activity remains the same in a period. Technological coefficients may vary over time.
4. The capacity of each facility, storage-area, is finite.
5. As the facilities may have different patterns of preventive maintenance schedules, the capacity of the machines may vary over a period of time.
6. The model allows any number of scenarios to occur. Each parameter of the model may vary in each scenario.
7. The sum of the probability associated with the scenarios should be equal to one.

8.4.1.3 Optimization Steps

The model and the DSS follow a set of activities in order to perform the optimization (Fig. 8.1). The production parameters and capacity limits are stored in the database tables [PROCESS.DAT]. A programming procedure generates the model files [Variables], [Constraints], and [Coefficients]. The list of variables is generated and stored in the [Variables] table. The list of model constraints is generated and stored in the [Constraints] table. The technological coefficients (model parameters) are generated and stored in the [Coefficients] table. A matrix generator (MG) program is designed to generate a mathematical programming system (MPS) file. The MPS is a text file, which is compatible with most optimizers including CPLEX, MINOS, etc. The matrix generator takes the input from the model files to generate an MPS file. The MPS file [PROBLEM.txt] is sent to the optimizers as an input to the model [PROCESS.mod]. The optimizer solves the model and returns the optimal solution [SOLUTION.txt]. A report writer program is designed to read the optimal solution and store the solution to the appropriate fields of the tables of the database.

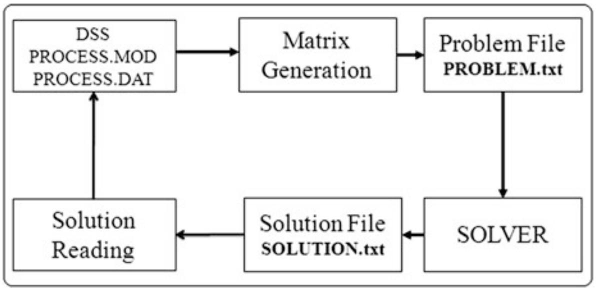


Fig. 8.1 Optimization steps

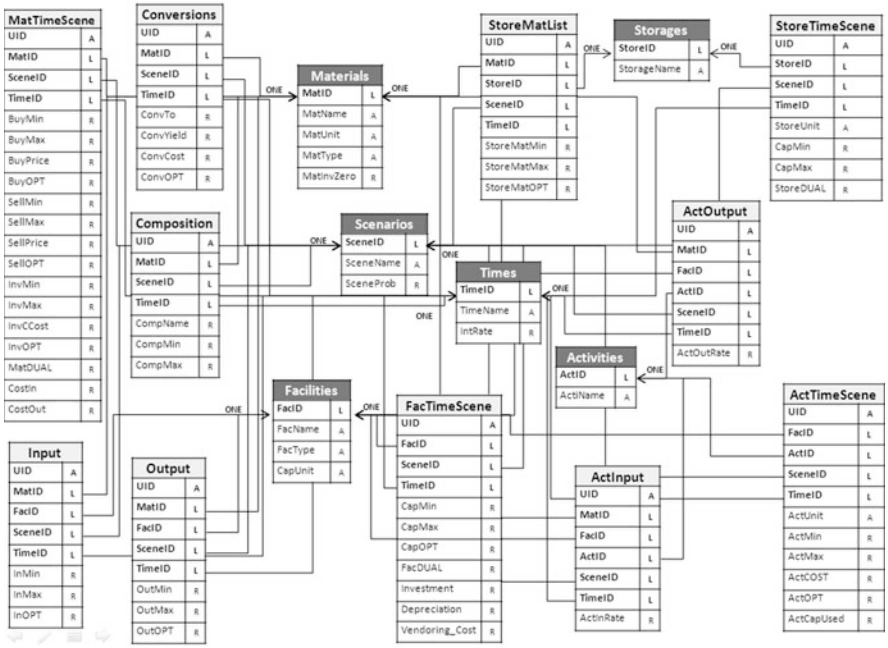


Fig. 8.2 Optimization model database structure

8.4.2 Database Structure

The database structure (Fig. 8.2) represents the SLP in terms of related objects like materials, facilities, activities and storages. The database has four model files and multiple data files. The database structure of the planning model is implemented in 4th Dimension, a Relational Database Management System of Adams and Beckett [1].

A set of diagnostic rules was designed and implemented in the database. The diagnostic rules were designed in order to ensure the correct entry of the data required to operate the optimization model. The rules were implemented in the form of procedures which validate data in the background. Each diagnostic runs in real time and ensures a correct record of the data before it is saved in the database files. Some examples of the rules are: the upper bound on variables should not be smaller than the lower bound; no more than one non zero for each pair of a variable and a constraint; the number of records in a file should not be more than the Cartesian product of the indexes of the file; the sum of probabilities associated with all scenarios should be one. As a default value, the lower limit for all variables is assumed to be zero, and the upper limit is assumed to be 99, 999, 999, 999. The default value of the yield is 100 % and that of the rolling rate is 1 ton per hour.

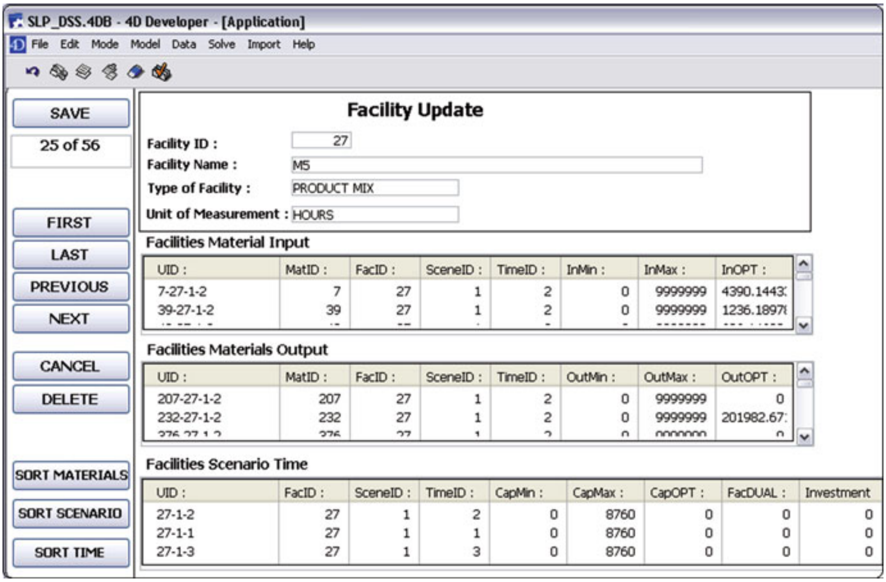


Fig. 8.3 Time scenario dependent included layout of facilities file

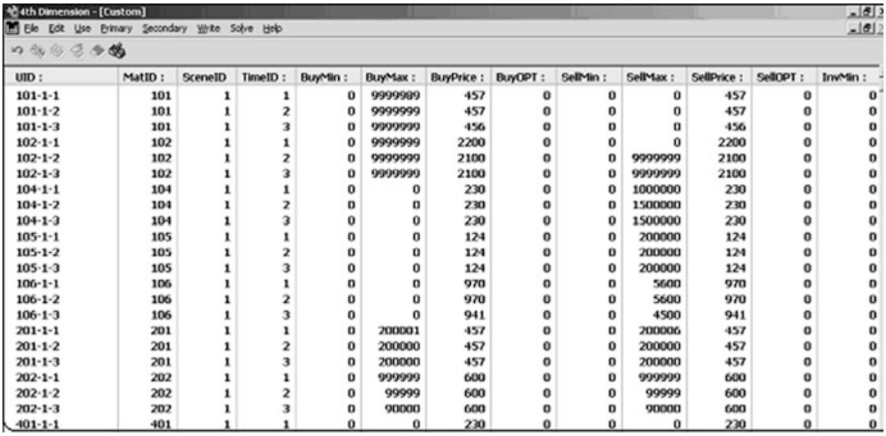
8.4.3 User Interface Development Experience

One of the major goals of this research was to develop a user-friendly interface so that end-users can operate the DSS with no or little knowledge of operations research/management sciences. The user-friendly screens facilitate the users to run the optimization steps and do the product mix optimization.

This DSS is designed to operate in three modes namely *Data*, *Update* and *Optimal*. The DSS is completely menu driven. Using these menus, users can switch to different modes. We describe each mode using an illustrative example. The *Data* mode is designed to enter the model parameters. Users input the data using a structured layout (Fig. 8.3). The entered model data can be displayed using a list layout (Fig. 8.4).

The *Update* mode is designed in order to update the parameters and simultaneously the model files such as the [Variables], [Constraints] and [Coefficients] files. The design facility of the *Update* mode enables users not to re-generate the model files from scratch but to update them, and thereby saves a lot of execution time. A screen illustrating the *Update* mode is shown in Fig. 8.5.

The *Optimal* mode is designed to display the multi-dimensional data and the optimal solution. In the *Optimal* mode, the optimal solution is reported on the screen (Fig. 8.6). A large number of screens are shared among different modes of the DSS with minor real-time customization. The DSS facilitates the display of the detailed summary of the model with each scenario and each time period (Figs. 8.7, 8.8).



UID :	MatID :	SceneID	TimeID :	BuyMin :	BuyMax :	BuyPrice :	BuyOPT :	SellMin :	SellMax :	SellPrice :	SellOPT :	InvMin :
101-1-1	101	1	1	0	9999989	457	0	0	0	457	0	0
101-1-2	101	1	2	0	9999999	457	0	0	0	457	0	0
101-1-3	101	1	3	0	9999999	456	0	0	0	456	0	0
102-1-1	102	1	1	0	9999999	2200	0	0	0	2200	0	0
102-1-2	102	1	2	0	9999999	2100	0	0	9999999	2100	0	0
102-1-3	102	1	3	0	9999999	2100	0	0	9999999	2100	0	0
104-1-1	104	1	1	0	0	230	0	0	1000000	230	0	0
104-1-2	104	1	2	0	0	230	0	0	1500000	230	0	0
104-1-3	104	1	3	0	0	230	0	0	1500000	230	0	0
105-1-1	105	1	1	0	0	124	0	0	200000	124	0	0
105-1-2	105	1	2	0	0	124	0	0	200000	124	0	0
105-1-3	105	1	3	0	0	124	0	0	200000	124	0	0
106-1-1	106	1	1	0	0	970	0	0	5600	970	0	0
106-1-2	106	1	2	0	0	970	0	0	5600	970	0	0
106-1-3	106	1	3	0	0	941	0	0	4500	941	0	0
201-1-1	201	1	1	0	200001	457	0	0	200006	457	0	0
201-1-2	201	1	2	0	200000	457	0	0	200000	457	0	0
201-1-3	201	1	3	0	200000	457	0	0	200000	457	0	0
202-1-1	202	1	1	0	999999	600	0	0	999999	600	0	0
202-1-2	202	1	2	0	99999	600	0	0	99999	600	0	0
202-1-3	202	1	3	0	90000	600	0	0	90000	600	0	0
401-1-1	401	1	1	0	0	230	0	0	0	230	0	0

Fig. 8.4 Materials time scenario output layout

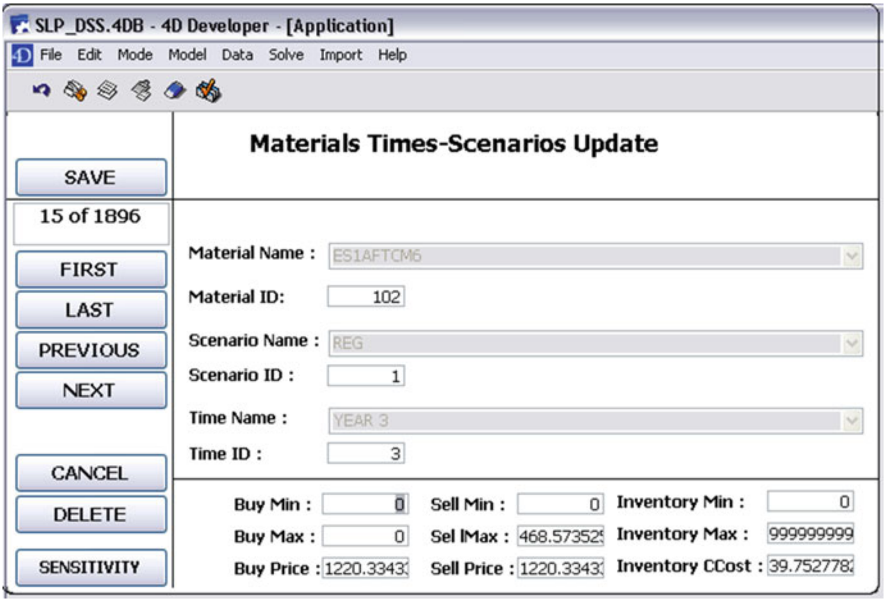


Fig. 8.5 Materials time scenario update layout

While developing the DSS, we had some difficulty explaining the concept to industry users, who were more used to Excel and or Solver. Very few could understand what exactly we were doing. There was a group of people who knew about spreadsheets and another who knew mathematical programming, but there were very few who were good at both. The very fact that we were merging both concepts had very little acceptance.

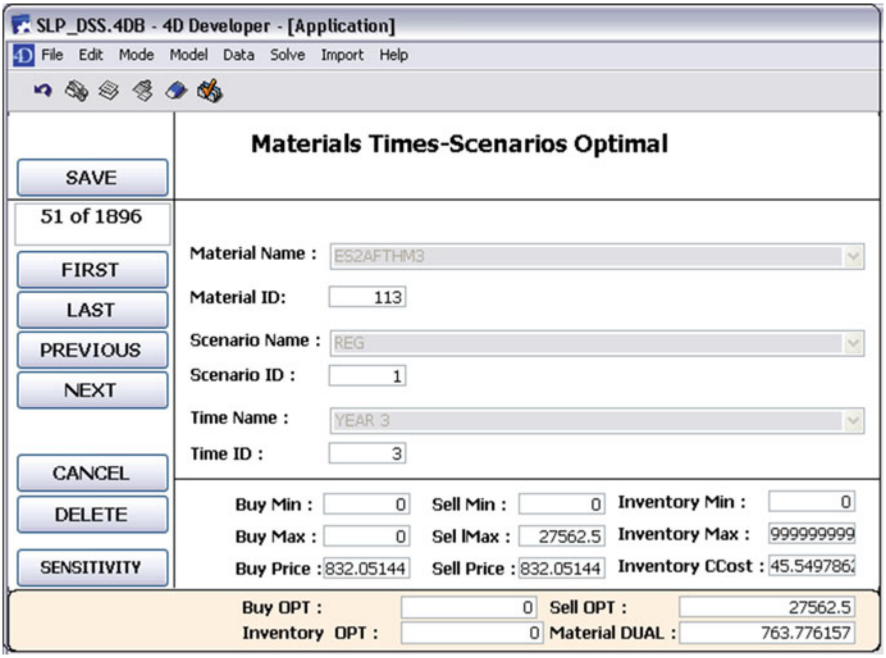


Fig. 8.6 Optimal solution reporting layout of material time scenario layout

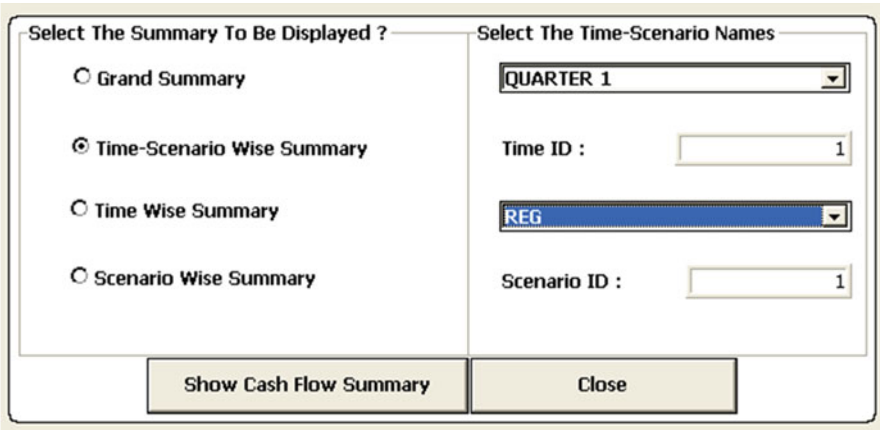


Fig. 8.7 Summary reporting option layout

It was difficult to create the *Update* mode, especially in the search and sort routine. These routines are difficult to implement as you need to search on four indices and then replace the value in the [Variables] file. The other difficulty was using the MPS format in CPLEX. Although Fourer [9] provided the super sparse

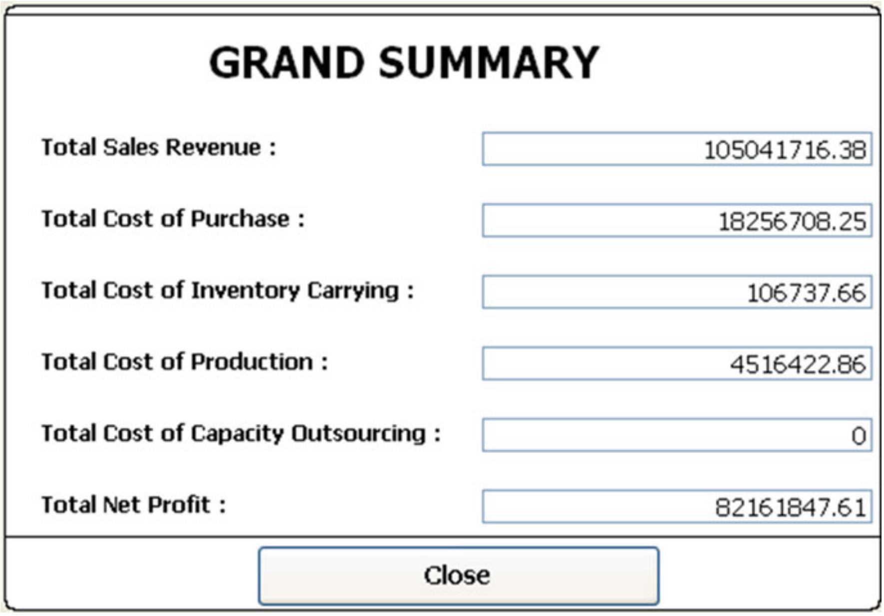


Fig. 8.8 Detailed summary of cash flow

matrix representation that is fast, we selected the MPS format as it is the industry standard. Similarly, we had difficulties in developing an included layout (Fig. 8.3).

8.4.4 Model and DSS Validation

The DSS was validated in order to test that the DSS produces meaningful results. The validation is also done to avoid the erroneous data entry into the DSS. An erroneous data entry in the DSS can cause errors in the optimization process. The validations of the model and the DSS are done in two different ways.

Firstly, we design a set of diagnostic rules. These diagnostic rules were designed and implemented in the DSS immediately before the entry of any new data. These rules ensure that incorrect entries can be restricted to the data entry level. The rules are designed in order to validate the data in line with the model requirements, and to avoid the potential infeasibility that may occur during optimization. The key causes of such infeasibilities are the illogical data entry. For example, the upper bounds of all the decision variables should not be less than the lower bounds. The sum of the probability of all scenarios should be equal to one. The number of records in a table should be equal to or less than the Cartesian product of the indices of each table. The detailed description of such rules are presented in Gupta et al. [14].

Secondly, we validated these diagnostic rules by entering erroneous data. We validated if the rules implemented in the DSS are functioning. The DSS was operated multiple times with the most possible combinations of erroneous data inputs and each diagnostic rule was tested for its validity.

Thirdly, the DSS was validated for meaningful optimization results. The DSS was tested to ensure that the model was correctly represented in the DSS, and was resulting in correct results. We subjected the DSS to extreme inputs in order to validate that the results were in line with the expected output. For example, the optimal solution should be zero for all decision variables when all the input parameters are zero. The optimal solution should be unbounded when there is no limit on the upper bound and no capacity constraints. The model should produce no finished products when the upper bounds on facilities, or finished products to be sold, or raw materials to be purchased, etc., are zero. A number of boundary tests were conducted to ensure the correctness of the optimization results.

8.5 Application of the DSS to a Pharmaceutical Company

In this section, we describe the application of the SLP model and the DSS to a pharmaceutical company. We design a set of experiments and test the DSS. We first describe the experiments, and then discuss the results. The database structure and the mathematical model of the SLP behind the DSS have been discussed in detail in our earlier publications [14].

The pharmaceutical company is located in India. The company is engaged in the business of production and marketing of tablets, injections and syrups. In this study, we consider only the case of tablets.

8.5.1 The Scale and Scope of Optimization

The scale of operations and the scope of optimization in tablet production are depicted in Table 8.1. The annual turnover of the company is USD 200 million with an annual production of $10,300$ tons. The large scale optimization is evident from the number of variables, constraints and coefficients. The scope of optimization is measured in terms of the variability in the key production parameters such as sell price, market demand for finished products, rate of production of each material in terms of the facility-activity ratio and the cost of operating an activity on a facility. We explain the ratios as a measure of scope of optimization with the help of an example. The sell price ratio is the ratio of the highest sell price of a finished product to the lowest sell price of another finished product among all finished products of tablets.

Table 8.1 Industry characteristics and optimization variability

Scale of operations	
Annual turnover (Million USD)	200
Annual production (Tons)	10,300.00
Size of the model	
Number of variables	3411
Number of constraints	3638
Number of coefficients (non zeros)	11,739
Sparseness (LP density non zeros)	0.09 %
Number of materials	76
Number of facilities (Machines)	22
Number of Activities (Transformations)	53
Number of planning periods	3
Number of scenarios	3
Scale of variability (Scope of optimization)	
Ratio of sell price (USD/Ton)	50.81
Ratio of market demand (Ton per annum)	4.3
Ratio of facility activity ratio (Ton/h)	526.03
Ratio of activity cost (USD /Ton)	1191.00

8.5.2 *The Process Flow of Tablet Production*

The process flow shows (Fig. 8.9) the typical production process of a tablet P1 (the actual names of the tablets have been disguised in order to ensure confidentiality). The production network of all finished products may have several such process flow sequences made up by a combination of machines. It is important to note here that the facility M1 appears twice in the process, first at the starting stage, where it gets three input raw materials and second, after the M7 facility, where it gets two input materials. One is the granulated mixture produced from M7 and the other is the raw material lubricant as an input to M1. Here, we see that the two stages of finished product P1 are competing to use the limited capacity of facility M1. This can be incorporated in the model by defining two different activities at facility M1 for the different processing stages. The complete network of facilities in the production process may have several such (limited capacity) competing situations. This operational complexity leads to the impact of optimization.

8.5.3 *Stochastic Optimization and Scenario Experiments*

The DSS was designed and developed for large scale optimization and modeling uncertainty in the model parameters. Since the model and DSS are generic, they can model uncertainty in any number of model parameters simultaneously. The

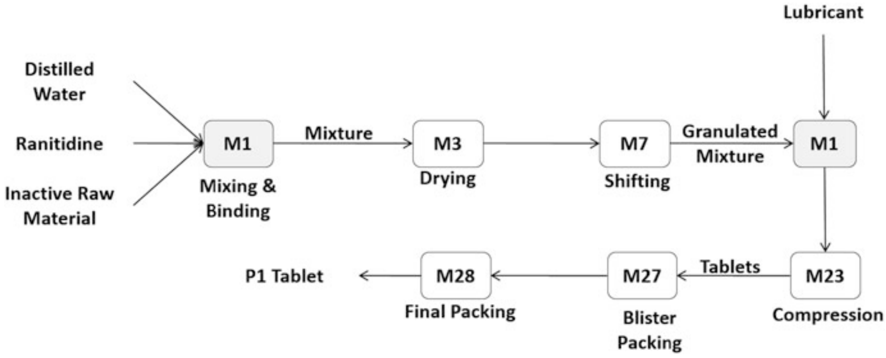


Fig. 8.9 A process flow diagram of a typical tablet production

emphasis in this chapter will remain on modeling uncertainty in the demand of tablets as finished products.

8.5.3.1 Stochastic Optimization Model

A stochastic optimization model considers hypothetical scenarios of occurrences of all the parameters of the model in association with their likelihood occurrence of probability. In order to understand the multi-scenario optimization model, a multi-period deterministic optimization model can be considered as a single-scenario model. In this chapter, we assume three scenarios in the model namely low demand scenario (L), regular demand scenario (R) and high demand scenario (H). These scenarios are based on the anticipation of the economic situation of geography. We vary the demand vector [D] of all finished products (tablets) in each scenario keeping all other model parameters constant. The demand vector [D] of the deterministic model is used in scenario R. We assume [0.8D] and [1.2D] as the demand vectors for scenario L and H, respectively. This hypothetical design of demand vectors is named as 20 % demand variability. Each scenario may have a different probability of occurrence such as equally likely (i.e., $1/3$ each), or left skewed (low probability to scenario L), or right skewed (low probability to scenario H).

8.5.3.2 Variants of the Model

Some variants of the model are the following:

Stochastic Linear Program (SLP): An SLP is a deterministic equivalent linear program (DELP). A DELP is solved considering all the scenarios together along with their probability. The objective of the DELP is the expected value of the net (contribution to) profit. Each constraint is replicated for each scenario outcome. The solution of the DELP is known as an SLP solution.

Perfect Information Solution (PI): A deterministic multi-period model is solved for each scenario separately. The expected value of the net (contribution to) profit of individual scenario model solutions is known as a PI solution. Mean Value Solution (MV): A deterministic multi-period model is solved using the expected value of the demand vector [D] for all finished products. The solution of this model is known as an MV solution.

Stochastic Freeze Solution (EEV): The decisions implemented before the realization of the uncertain demand are known as first stage decisions. The decision variables associated with such decisions are production quantity, facility capacity consumption, raw material quantity to purchase etc. In the EEV model the SLP model is resolved by freezing (fixing) the value of the first stage decision variables with the optimal solution of the SLP model solution. The stochastic freeze solution is also known as the expected value of expected solution (EEV).

Value of Stochastic Solution (VSS): The value of the stochastic solution is a common measure of the impact of modeling uncertainty using SLP. The VSS is determined as a difference of the optimal SLP solution and the EEV solution.

Improvement in NP: This is the percentage improvement in net (contribution to) profit from SLP over EEV. This is the value of VSS in terms of the percentage of the stochastic freeze solution.

Expected Value of Perfect Information (EVPI): This is determined as the difference between the PI solution and the SLP solution.

In order to determine the VSS and EVPI, one has to develop a total of six instances of the model and solve them. The PI solution needs three model instances, SLP is one instance, FSLP is another instance, and MV is the base case deterministic instance.

8.5.3.3 Stochastic Experiments Design

We designed a set of experiments. The design is adopted from Leung et al. [16]. We developed a total of nine cases to understand how the SLP solution changes. These nine cases are made of a Cartesian product of three demand variability cases and three discrete probability distribution cases. We assumed three demand variability cases—20, 30 and 40 %. The three probability distributions were equally likely (EL), left skewed (LS) and right skewed (RS). We developed and solved a total of 54 model instances in order to determine the VSS and EVPI ($6 \times 3 \times 3$). The six model instances are required for each case. The three discrete probability distributions can be seen in Table 8.2. The three demand variability cases can be observed in Table 8.3.

Table 8.2 The three cases based on probability distribution

Probability distribution cases	LOW	REG	HIGH
Right skewed	0.75	0.15	0.1
Equally likely	0.33	0.33	0.33
Left skewed	0.1	0.15	0.75

Table 8.3 The three cases based on demand variability

Demand variability cases	LOW	REG	HIGH
20 % Demand variability (Mean = D, SD = 0.027D)	80 % of [D]	[D]	120 % of [D]
30 % Demand variability (Mean = D, SD = 0.060D)	70 % of [D]	[D]	130 % of [D]
40 % Demand variability (Mean = D, SD = 0.107D)	60 % of [D]	[D]	140 % of [D]

8.6 Results: Analysis and Discussion

The optimization results of the nine cases and the six model instances of each model are presented in Table 8.4. We present the results of all the 54 model instances. The results of the 20 % demand variability and equally likely probability distribution are considered for the discussion. We find a few interesting observations from the results of optimization.

The impact of modeling the uncertainty is visible from the VSS as USD 247, 621. It is equivalent to 1.05 % net (contribution to) profit compared to the EEV solution. Managers follow the MV solution, but due to their ignorance of the demand distribution information, they end up receiving only the EEV solution. The EEV is an inferior solution compared to the SLP solution. The VSS is the measure of an impact of using the probabilistic demand distribution information into the LP model.

It is important to note that the VSS is USD 247, 621, which can be achieved by incorporating the additional demand distribution information into the SLP model. The cost of buying this information is substantially lower than the VSS. This demonstrates the impact of modeling uncertainty using SLP.

Another interesting observation is the value of EVPI as USD 6647. The EVPI can be achieved with the availability of perfect information, which may be either very expensive or impossible to achieve. The value of EVPI is substantially smaller than the cost of perfect information. This further demonstrates that it is worth buying partial information and using that in the SLP as compared to buying perfect information.

The optimization results of each of the nine cases show an interesting pattern. We observe a decreasing pattern of model solution (20 %D, and EL) in the following sequence $[(Z_{MV} = 24.65) \geq (Z_{PI} = 23.74) \geq (Z_{SLP} = 23.73) \geq (Z_{EEV} = 23.48)]$. This pattern is in line with the SLP literature [2]. This pattern is consistent across all the nine cases in Table 8.4.

Table 8.4 Pharmaceutical: results of experiments from multi-scenario planning

Demand variability skewness		Unit	Right skewed	Equal likely	Left skewed
20 %	Perfect information (PI)	USD Mn	22.23	23.74	24.74
	Mean value solution (MV)	USD Mn	22.51	24.65	25.01
	Stochastic solution (SLP)	USD Mn	22.23	23.73	24.72
	Stochastic freeze (EEV)	USD Mn	22.03	23.48	24.09
	(VSS)	USD	195,025	247,621	630,613
	% Improvement in NP	Percentage	0.89 %	1.05 %	2.62 %
	EVPI	USD	2025	6647	14,586
30 %	Perfect information (PI)	USD Mn	21.02	23.28	24.78
	Mean value solution (MV)	USD Mn	21.43	24.65	25.19
	Stochastic solution (SLP)	USD Mn	21.02	23.27	24.76
	Stochastic freeze (FSLP)	USD Mn	20.73	22.9	23.82
	VSS	USD	292,538	371,277	945,878
	% Improvement in NP	Percentage	1.41 %	1.62 %	3.97 %
	EVPI	USD	3036	9968	21,876
40 %	Perfect information (PI)	USD Mn	19.81	22.82	24.83
	Mean value solution (MV)	USD Mn	20.36	24.65	25.38
	Stochastic solution (SLP)	USD Mn	19.81	22.81	24.8
	Stochastic freeze (FSLP)	USD Mn	19.42	22.31	23.54
	VSS	USD	390,050	494,933	1,261,142
	% Improvement in NP	Percentage	2.01 %	2.22 %	5.36 %
	EVPI	USD	4048	13,289	29,166

8.7 User Experiences

We describe the lessons that we learnt in our journey of design, development, implementation and testing of the optimization-based DSS.

8.7.1 Lessons from the End User Perspective

There are a number of challenges in the development and implementation of a DSS from the end-user point of view. Though the design and development should be top management initiatives, the end-users should be taken into consideration before such projects begun. The success of the project lies in involving all the stakeholders of the project at all stages of the project. This participation can be achieved through a series of awareness workshops across the organization. Resolving end-user queries helps users to understand the need for such systems and facilitates easy implementation.

Industry practitioners should be presented with a prototype of the DSS in order to build confidence in the model-based DSS, OR/MS and technology driven tools. The managers and end-users need to be involved in the design and development of the

DSS on an iterative basis. This is important in order to ensure that the organizational change will be accepted by the practicing managers.

Managers may move to different positions and different locations in the industry. As a result, it is very difficult to convince new managers about the decisions made by previous managers. A smooth and involved transition can ensure continued support from the new managers.

The biggest challenge in the implementation and use of a DSS is the acceptance of the change, especially in the way that managers operate their day to day functions. It is important to apprise managers and end-users, how they would benefit individually in the implementation and use of a DSS. Managers want to know if the DSS will save their time, effort and or money. They also want to know that it is not a threat to their job. The idea of a DSS should be promoted on the basis of individual benefits, which in turn leads to organizational benefits. It is important to share how the DSS is different and better than their existing system.

The optimization-based DSS for single-period planning by Fourer [10] and Dutta et al. [5] and Dutta and Fourer [4] was successfully implemented in a steel company in North America. The need for the DSS was initiated by the company. The managers and end-users were involved in the design and development of the DSS. The DSS also demonstrated a significant impact on the bottom line of the company. The multi-period planning DSS by Dutta et al. [4] and multiple scenario planning DSS by Gupta et al. [14] were designed and developed in consultation with industry managers.

The DSS could not be implemented due to a change in the management. It became difficult to convince the new management that the implementation and use of the multiple period, multiple scenario planning model was a value proposition. The new management did not show much interest in this DSS due to their changed focus and lack of appreciation for the technology and OR/MS tools.

8.7.2 Key Characteristics of a Good Model-Based DSS

Based on our review of the literature and our experience of the design and development of a multiple period optimization-based DSS, we describe the characteristics of a good model-based DSS. The DSS should be generic in nature. It should have model, data and solver independence. It should enable the generation of any number of instances from a new set of data. The model can be a mathematical model, simulation model, or any other OR/MS model. It should be designed and developed as the heart of the DSS. The data instance should be combined with the model and provided as an input to the solver separately.

The DSS should be menu driven and very user-friendly. The managers and end-users may have little or no understanding of the models of OR/MS and database management. Practitioners perceive the DSS based on OR/MS models as an academic exercise. It is important to design and develop an interface which can help the end-users to use the system.

Most DSS fail at the implementation stage due to a lack of user-friendly interface. It is important to note that data entry should be made only once. Multiple entries of the same data into the tables may lead to errors in optimization. The database should allow auto generation of the indices of the tables. The indices of each table help a quick processing of the data. The selection of the static data related to the model schema for entering the model instance data should be made using the drop down menus. The DSS should be capable of receiving data using multiple methods, for example, data entry using the input screens, import of bulk data using a delimited text file, import of the model schema or model instance from existing model files, etc.

A DSS should facilitate the display of all data-related and solution-related reports that users may need for their decision making. It should facilitate the import of an optimal solution from an optimization model and the export of a solution to the required files. Multi-dimensional data should be reported using included layouts. Data with multiple indices is known as multi-dimensional data, for example, the units of a material to be used as an input in an activity of a facility in a period, in a scenario.

The DSS should be equipped with the facility for scenario analysis. Users may like to conduct a scenario analysis (sensitivity analysis) in the DSS and verify the behavior of the solution in multiple scenarios. It should also facilitate multiple-period planning. The time required to run the DSS in order to receive the optimal solution should be reasonably low.

8.7.3 Challenges Addressed in Model-Based DSS

In our DSS we demonstrated the model, data and solver independence. The SLP model was the base of the optimization-based DSS. The SLP was modeled and represented in the form of a relational database structure. The database structure had a one to one correspondence between the fields of the tables of the database with the decision variables and the parameters of the SLP model. The SLP was also represented in the form of a matrix generation (MG) procedure. The procedure was written to develop an MPS file of the model instance. The MPS file is also an MG form of the SLP representation.

The model designed in the form of a database structure and MG procedure is independent of the data model instance. The data model instance is in the form of fields of the tables of the database structure, the MPS file and the model files format. The model files are the [Constraints], [Variables] and [Coefficients] tables of the database. We used MINOS and CPLEX as the solvers. The DSS is also generic and allows planning using any process industry data instance. The model and DSS can be used for any number of materials, facilities, activities, storage areas, times and scenarios. This DSS is user-friendly, menu-driven and requires little or no knowledge of OR/ MS. Our DSS demonstrated multiple methods of entering data. The model schema data entered into the DSS remain unique and consistent across all screens. We demonstrated the facility of solution reporting, multiple period planning, multiple scenario planning, sensitivity analysis, etc. The DSS takes very little time to generate the model instance and solve it.

The multiple period planning facility allows users to undertake strategic, tactical and operational planning by changing the planning horizons. It also accounts for the cost of capital. The DSS is customized to solve an SLP with discounted objective functions. The key strategic planning decisions supported by our DSS were optimal product mix, capacity expansion decisions, capacity outsourcing decisions, product diversification decisions, capital budgeting decision, etc.

8.8 Conclusions

In this chapter, we described the design, development, implementation and testing of an optimization-based DSS developed by Fourer [10], Dutta et al. [4], and Gupta et al. [14]. We discussed the database structure of the DSS, the fundamental elements of the process industry model and the testing of the DSS using a set of real data from a pharmaceutical company. We demonstrated how multiple scenario, multiple period planning can be done using a generic, menu-driven, user-friendly DSS. Managers need little or no knowledge of OR/MS to use the DSS and do the optimization.

The reported applications of SLP in process industry production planning are very limited. We demonstrated that the uncertainty in model parameters can be modeled using the SLP in a user-friendly manner. We designed and conducted a set of experiments on the model and the DSS. The results of the optimization were discussed to demonstrate the impact of optimization and modeling uncertainty. We were able to demonstrate that VSS is significant and it is worth using SLP for modeling uncertainty.

We summarized the chapter by discussing the lessons that we learnt in our journey of the development and implementation of the DSS. We described the lessons that we learnt from the end-user point of view and from the modeler's point of view. We also discussed the important characteristics of a good model-based DSS. The features of our DSS were also discussed in the same context. We conclude the chapter by stating that managing organizational change and developing a user-friendly interface can lead to the success of a model-based DSS. The journey of studying the reported literature on model-based DSS, design and development of the optimization-based DSS, and the testing and the implementation phases of the DSS in industry has been most enriching and exciting.

References

1. Adams, D., Beckett, D.: Programming 4th Dimension: The Ultimate Guide. Foresight Technology Inc., Washington, DC (1998)
2. Birge, J.R.: State-of-the-art-survey-stochastic programming: computation and applications. INFORMS J. Comput. **9**(2), 111–133 (1997)
3. Dantzig, G.B.: Linear programming under uncertainty. Manag. Sci. **1**(3–4), 197–206 (1955)
4. Dutta, G., Fourer, R.: An optimization-based decision support system for strategic and operational planning in process industries. Optim. Eng. **5**(3), 295–314 (2004)
5. Dutta, G., Sinha, G.P., Roy, P., Mitter, N.: A linear programming model for distribution of electrical energy in a steel plant. Int. Trans. Oper. Res. **1**(1), 17–29 (1994)
6. Dutta, G., Fourer, R., Majumdar, A., Dutta, D.: An optimization-based decision support system for strategic planning in a process industry: the case of a pharmaceutical company in India. Int. J. Prod. Econ. **106**(1), 92–103 (2007)
7. Dutta, G., Gupta, N., Fourer, R.: An optimization-based decision support system for strategic planning in a process industry: the case of aluminium company in India. J. Oper. Res. Soc. **62**(4), 616–626 (2011)
8. Fazlollahtabar, H., Asadinejad, M., Shirazi, B., Mahdavi, I.: Proposing a decision support system for service oriented manufacturing based on enterprise resource planning. J. Inf. Technol. Softw. Eng. **3**(1), 1–7 (2013)
9. Fourer, R.: Modeling languages versus matrix generators for linear programming. ACM Trans. Math. Softw. **9**(2), 143–183 (1983)
10. Fourer, R.: Database structures for mathematical programming models. Decis. Support. Syst. **20**(4), 317–344 (1997)
11. Fragniere, E., Gondzio, J.: Stochastic programming from modeling languages. In: Wallace, S.W., Ziemba, W.T. (eds.) Applications of Stochastic Programming. SIAM Series on Optimization, pp. 1–21. SIAM, Philadelphia (2002)
12. Galasso, F., Mercé, C., Grabot, B.: Decision support for supply chain planning under uncertainty. Int. J. Syst. Sci. **39**(7), 667–675 (2008)
13. Gupta, V., Peters, E., Miller, T., Blyden, K.: Implementing a distribution-network decision-support system at Pfizer/Warner-Lambert. Interfaces **32**(4), 28–45 (2002)
14. Gupta, N., Dutta, G., Fourer, R.: An expanded database structure for a class of multi-period, stochastic mathematical programming models for process industries. Decis. Support. Syst. **64**, 43–56 (2014)
15. Hamamoto, S.: Development and validation of genetic algorithm-based facility layout a case study in the pharmaceutical industry. Int. J. Prod. Res. **37**(4), 749–768 (1999)
16. Leung, S., Wu, Y., Lai, K.: A stochastic programming approach for multi-site aggregate production planning. J. Oper. Res. Soc. **57**(2), 123–132 (2006)
17. Mansouri, S.A., Gallear, D., Askariazad, M.H.: Decision support for build-to-order supply chain management through multiobjective optimization. Int. J. Prod. Econ. **135**(1), 24–36 (2012)
18. Miller, T., Matta, R.: Integrating production and transportation scheduling decisions between two geographically separated plants. J. Bus. Logist. **24**(1), 111–146 (2003)
19. Miller, T., Peters, E., Gupta, V., Bode, O.: A logistics deployment decision support system at pfizer. Ann. Oper. Res. **203**(1), 81–99 (2013)
20. Palanisamy, S., Palaniappan, P., Patel, N.A., Dinesh, S., Adeni, M.R.F.: Decision support system for flow line assembly planning in an automobile manufacturing company. Int. J. Appl. Eng. Res. **11**(6), 4325–4330 (2016)
21. Salam, M.A., Sami, A.K.: Simulation based decision support system for optimization: a case of thai logistics service provider. Ind. Manag. Data Syst. **116**(2), 236–254 (2016)
22. Scott, J., Ho, W., Dey, P.K., Talluri, S.: A decision support system for supplier selection and order allocation in stochastic, multi-stakeholder and multi-criteria environments. Int. J. Prod. Econ. **166**, 226–237 (2015)

23. Varlan, E., Le Paillier, R., et al.: Multicriteria decision making for contract research organisation choice in the pharmaceutical industry. J. Oper. Res. Soc. **50**(9), 943–948 (1999)
24. Venkatadri, U., Srinivasan, A., Montreuil, B., Saraswat, A.: Optimization-based decision support for order promising in supply chain networks. Int. J. Prod. Econ. **103**(1), 117–130 (2006)
25. Viswanadham, N., Narahari, Y.: Queueing network modelling and lead time compression of pharmaceutical drug development. Int. J. Prod. Econ. **39**(2), 395–412 (2001)
26. Wallach, D., Loisel, P.: Effect of parameter estimation on fertilizer optimization. Appl. Stat. **43**(4), 641–651 (1994)
27. Xue, F., Sanderson, A.C., Graves, R.J.: Multiobjective evolutionary decision support for design–supplier–manufacturing planning. IEEE Trans. Syst. Man Cybern. Part A Syst. Humans **39**(2), 309–320 (2009)

Chapter 9
Decision Support in Water Resources Planning and Management: The Nile Basin Decision Support System

Andreja Jonoski and Abdulkarim H. Seid

Abstract This book chapter presents the experiences in developing and applying a decision support system in a trans-boundary river basin context taking the Nile Basin Decision Support System (NB DSS) as a case. This system is categorized as *model-based DSS*, as simulation models of the water resource system are its central components, albeit combined with many other components such as a database, Geographical Information Systems (GIS), and algorithm implementations for data analysis, optimization, Multi Criteria Analysis (MCA), Cost Benefit Analysis (CBA), etc. The system is developed cooperatively, as part of the efforts of the Nile Basin Initiative (NBI) for promoting shared and equitable water resources development and management in the basin. It serves its formal purpose of evaluating different water resources planning and management alternatives, but to also acts as knowledge sharing platform for enhancing shared understanding of the Nile basin water resources system and the interests and needs of various stakeholders. The chapter presents the process for NB DSS design, development and implementation, together with its main components and the experiences and lessons learned from its initial use.

9.1 Introduction

The field of Water Resources Planning and Management (WRPM) is characterized with complex decision making problems in which possible alternatives (options) as well as the objectives (criteria) on which they need to be evaluated depend on many factors. Firstly, the physical system providing the water resources (e.g. a

A. Jonoski (✉)
UNESCO-IHE Institute for Water Education, PO Box 3015, 2601 DA Delft, The Netherlands
e-mail: a.jonoski@unesco-ihe.org

A.H. Seid
NBI Secretariat, P.O. Box 192, Entebbe, Uganda
e-mail: aseid@nilebasin.org

© Springer International Publishing Switzerland 2016
J. Papathanasiou et al. (eds.), *Real-World Decision Support Systems*,
Integrated Series in Information Systems 37,
DOI 10.1007/978-3-319-43916-7_9

river basin) has its own complexity due to the dynamic and non-linear interactions existing within the natural hydrological cycle, often altered by human interventions. Secondly, given that water is one of the most shared resources, indispensable for sustaining nature as well as numerous socio-economic activities, decisions regarding water resources are always with multiple objectives and frequently involve multiple decision makers and stakeholders. Finally, the way in which these diverse decision makers and stakeholders are organized from institutional and administrative point of view, conditions the structuring of WRPM problems. This implies that the analysis and development of Decision Support Systems (DSSs) in this field need to take into account the interactions among three systems: the natural-physical system, the socio-economic system and the institutional-administrative system [16]. Most such DSSs, however, are primarily focused on the natural-physical system (e.g. river basin), aiming at exploring, designing and proposing alternatives that constitute interventions in this physical system (e.g., introducing a water intake for urban water supply, building and operating a multi-purpose reservoir, constructing flood protection levees, developing a new irrigation area for agricultural production, etc.). Analysis of the socio-economic system then provides the main objectives (and their corresponding indicators) for evaluating these different alternatives. The institutional-administrative system provides the decision making context and drives the development and use of DSSs in water resources.

Formally, the problems that DSSs for WRPM address can be summarized as follows:

1. Formulate a set of objectives that guides certain water resources management project or plan. These are commonly defined using indicator values that need to be maximized (e.g. area of irrigated agricultural land, hydropower production etc.) or minimized (e.g. costs, flood damage, etc.).
2. Explore, design and propose a set of alternatives that can be implemented for meeting the selected objectives. These can be defined using decision variables that are varied with the proposed alternatives in the water resources system, or by combining a set of heterogeneous lower level interventions (often called 'measures') into alternatives to be considered.
3. Evaluate and rank the proposed alternatives by employing some decision evaluation methods. The primary task here is to establish relations between the proposed alternatives and objectives, which can be used to evaluate the performance of each alternative with respect to the objectives. This is most commonly a simulation model, which needs to have appropriate representation of the elements of the water resources systems depending on the actual decision problem at hand. Such a model can then be combined with other evaluation methods to provide final ranking of the proposed alternatives.

DSSs in WRPM are usually built on a scale of a river basin. Two important characteristics of water resources in a river basin are their limited availability (finiteness) and their spatio-temporal variation. This means that water users in the basin need to share the same resource, preferably in equitable fashion that avoids competition and conflicts. This requires taking into account the upstream-downstream and seasonal variations of water resources availability, leading to complex interactions among water users distributed over the basin.

To capture these interactions, DSSs in WRPM rely critically on simulation models. Commonly, these simulation models are called ‘river basin simulation models’ which simulate alternatives of water allocation to different users, given storage (e.g. reservoirs), distribution (e.g. canals, rivers) and user (e.g. irrigation area) elements of the existing or planned water infrastructure. Examples of such models are RIBASIM [6], MIKE BASIN [7], WEAP [24] and MODSIM [3] (see also for comparative analysis of such models [26]). Since actual water resources availability depends on precipitation, these models are often combined with hydrological (rainfall-runoff) models. When spatio-temporal variations of flows and river water levels are needed, hydraulic models are used. In any case, specialized simulation models are critical components of any DSS for WRPM. Therefore, using the DSS classification introduced in the introductory chapter of this book [23], DSSs in WRPM fall in the category of *model-driven DSSs*.

Simulation models alone however, do not constitute a DSS in WRPM and a number of additional components are needed. Any DSS for WRPM critically depends on available data, which are used for setting-up, calibrating and validating the simulation models, as well for separate data analyses, irrespective of the models. Sources of such data can be in situ monitoring networks, remote sensing, project-dependent measurement campaigns, etc. Data management components constitute the so-called information layer of DSSs in WRPM. With many spatially distributed data Geographical Information Systems (GIS) play an important role in these DSSs, especially in their user interfaces. Outputs of simulation models need to be converted into decision relevant metrics (indicators) for use in evaluation tools such as optimization, MCA or CBA. Such components provide better understanding of consequences of implementing various alternatives under different set of climatic and socio-economic settings. When bringing all these components together, a generic structure of a DSS for WRPM may be introduced as in Fig. 9.1.

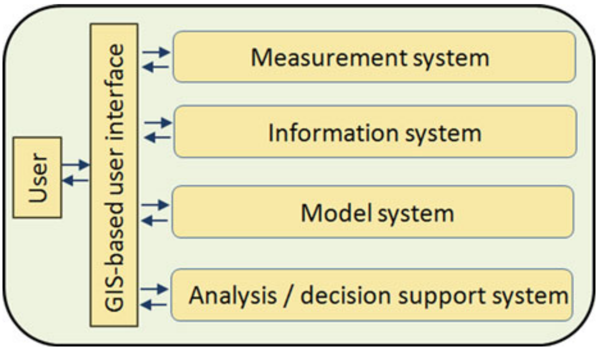


Fig. 9.1 Generic components of a DSS for WRPM (adapted from [16])

Most DSSs in WRPM developed over the last two decades have elements of this generic structure (e.g. [2, 12, 18, 22, 28], see also [17] for broader environmental DSSs). Parallel to the technological progress in developing such systems, the WRPM community has analyzed the critical issues that such systems need to address and the challenges for their more widespread use in real decision support situations [4, 15, 25]. Recognized issues are the need for integrated systems that enable analysis of all water-resources aspects in the basin relevant to the particular decision problem at hand, clear framework and decision support steps, use of appropriate data and models, and the critical need of involving end-users (stakeholders) in the process of developing the DSS. The need for flexible and extensible systems, increasingly web-based, has been stressed to meet the needs of diverse and distributed stakeholders [1, 5, 11, 13, 14]. An important requirement of DSSs for WRPM has been to actively support the process of decision making in which software developers and DSS experts support stakeholders and decision makers to develop shared understanding of the nature of the problems at hand, values and interests of others, and trade-offs involved in the proposed alternatives. This requirement has become even more important than the formal task of evaluating alternatives, as it contributes to the individual and social learning of the stakeholders, leading to better and more accepted decisions [21].

Meeting these requirements in developing countries is more challenging because of lack of data, low data quality, insufficient modeling capacity, lack of stakeholder participation and challenging political decision making contexts. Further complexities arise in trans-boundary river basins, where the limited water resources with large spatio-temporal variability need to be shared by different countries [9, 10]. The Nile River Basin is one such example, covering 11 African countries with different interests, upstream-downstream relations, different level of socio-economic development, and different languages and cultures. This chapter presents the experiences of developing and applying the Nile Basin Decision Support System (NB DSS), which has recently been introduced in the Nile countries. Following this introduction, Sect. 9.2 presents the main characteristics of the Nile basin and of the NB DSS. Section 9.3 presents the user and system requirements and Sects. 9.4 and 9.5 introduce the system design and implementation. Section 9.6 briefly presents the user interfaces followed by Sect. 9.7 with few use-cases of NB DSS. Future prospects for NB DSS are presented in Sect. 9.8, followed by the concluding Sect. 9.9.

9.2 Main Characteristics of the Nile Basin and the NB DSS

The Nile Basin covers an area of about 3.17×10^6 km^2 distributed over the countries: Burundi, DR Congo, Egypt, Eritrea, Ethiopia, Kenya, Rwanda, South Sudan, the Sudan, Tanzania and Uganda (Fig. 9.2).

Main challenges in the Nile basin are the limited availability of water resources and their high spatio-temporal variations. The average annual rainfall of about 650–

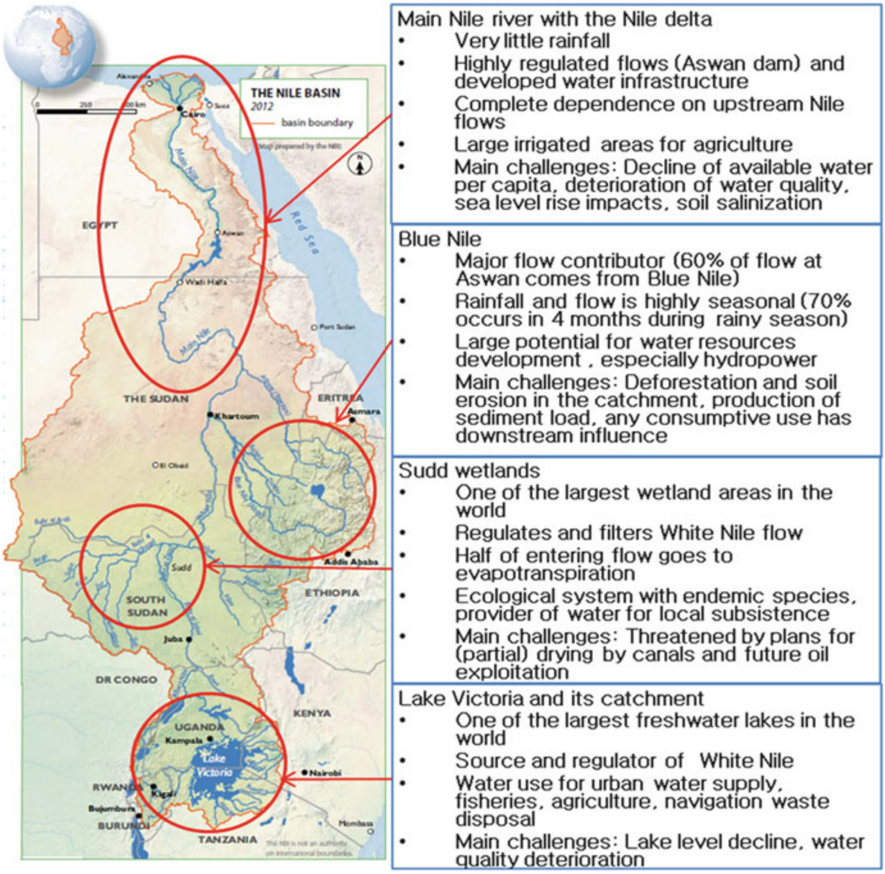


Fig. 9.2 The Nile Basin and its main characteristics (adapted from [19])

700 mm/year is unevenly spatially distributed. Most of it occurs in the upstream parts of Lake Victoria and Ethiopian Highlands, whereas the downstream parts of Sudan and Egypt receive very little rainfall. Moreover, the largest rainfall contributing area of the Blue Nile is highly seasonal in nature, with most of the rainfall occurring in the wet season (June-September). The average annual natural flow at Aswan for the period from 1900 to late 1950s was estimated to be about 84×10^9 m^3/year; more recent estimates of the flow show higher annual runoff values. However, compared to many large river basins in the world, the Nile has a relatively low yield; the runoff coefficient is on the order of 4 %. This limited amount of water resources needs to be shared in equitable manner across all countries of the basin.

Key water uses in the basin are irrigated agriculture for food security, hydropower development, navigation and sustaining valuable ecosystems. Historical water resources developments and agreements allocate significant amount of water resources to downstream countries (Egypt, Sudan), but upstream countries (Ethiopia and the countries around Lake Victoria) are eager to develop their own water resources (e.g. Ethiopia is currently constructing the Grand Ethiopia Renaissance Dam—GERD on the Blue Nile). Many decisions regarding water resources have taken place within individual countries, which creates high potential for conflicts and there is clear need for cooperation on the scale of the whole river basin.

Recognizing the importance of cooperatively managing their shared water resources, the Nile Basin countries, supported by the international community, have created the Nile Basin Initiative (NBI) as a regional inter-governmental partnership for information sharing, discussions and negotiations among all stakeholders regarding WRPM. One significant action taken by NBI was the development of the Nile Basin Decision Support System (NB DSS). The first version of this DSS was finalized in 2012.

In order to position NB DSS within the broader DSS research and practice, its key characteristics are presented using the DSS check list proposed by Power [23] in Table 9.1 below.

It needs to be noted that NB DSS is not the first attempt to provide decision support tools for managing WRPM in the basin. Previous developments for DSSs in WRPM for the Nile such as [8], however, were focused only on parts of the Nile Basin; the Blue Nile was not adequately covered by the model for lack of data at the time. This undertaking was not part of an NBI project, though it aimed at developing a decision support system for the Nile and it was implemented in close collaboration with Nile Basin countries. Drawing on experiences of the first attempt, NB DSS was developed by putting large emphasis and investments in stakeholder involvement and local capacity development during the system design, development and implementation, leading to broader acceptance and more widespread future use of the system. Current information on NB DSS and support for its users is available at [20].

Table 9.1 NB DSS categorization using the DSS check list of Power [23]

DSS check list	NB DSS
1. What is the dominant component of the architecture that provides functionality?	1. Suite of simulation modeling tools, thus this is primarily a *Model-based DSS*. Many other supporting components are included.
2. Who are the targeted users?	2. Water resources planners, managers and experts in Nile basin countries.
3. What is the purpose of the DSS?	3. Evaluating alternatives of water resources development and management schemes; understanding river system behavior; sharing knowledge; supporting decision making from regional perspective.
4. What enabling technology is used to deploy the DSS?	4. Existing water system simulation models combined with databases, GIS and algorithms for optimization, MCA and CBA.

9.3 Users and System Requirements

The Nile Basin DSS was envisaged as analytical tool for decision making in water resources planning and management by Nile riparian countries. Identification of the potential users was carried out under the DSS needs assessment and conceptual design phase, which focused on scoping the DSS through answering the following questions:

1. What are the key thematic focus areas for the Nile Basin DSS?
2. What are the decisions to be supported in these thematic areas?
3. What outputs should it produce to support these decisions?
4. Who are the main users and clients of the DSS?
5. How do the users interact with the system?
6. What data is available to support the DSS development and use?
7. How should future users be involved during development?

Users of NB DSS were broadly divided into two main categories. The first category comprises of users who may or may not run the DSS themselves but use the information generated by the DSS in their water resources planning and management related activities. Examples of such users are decision makers, members of the Nile Basin Initiative governance, senior water resources planners at national water ministries, range of stakeholders that have interest in the management and development of the Nile. Users in this category may not have background in modeling but know what information they need to address specific water resources issue they handle. The second category of users comprises of those users who interact with the DSS software to accomplish a certain task in their work routine. This category of users is interested in the tools the DSS provides, its features, and how the DSS can help them accomplish tasks such as modeling, time series analysis, tradeoff analysis, etc. This category of users was further divided into classes depending on the role they might assume in any organizational setup where the DSS is installed and used. This classification will be discussed later in this section together with user requirements.

Identification of user requirements started with identification of the thematic scope of the DSS and high level user requirements. In this stage, a range of potential users (category one) were engaged who identified spectrum of information that can be generated through NB DSS. The consultations with broad spectrum stakeholders led to formulation of an agreed common goal for NB DSS:

The ***goal*** *of the Nile Basin DSS is "to serve as shared knowledge base, provide common analytical capability, and support stakeholder interaction, for cooperative planning and management decision making for the shared Nile River Basin".*

Table 9.2 Thematic focus areas of Nile Basin DSS

1: **Water resources development**: deals with the planning decisions regarding water resources infrastructure development.	5: **Energy resources development (focus on hydropower)**: supporting planning decisions for the development of hydropower potential.
2: **Optimal water resources utilization**: focuses on planning decisions for enhancing efficient utilization of available water resources.	6: **Rain-fed and irrigated agriculture**: assessing current productivity and production levels of rain-fed and irrigated agriculture; supporting planning decisions of irrigated agriculture.
3: **Coping with flood**: main focus shall be to provide information on characteristics of flood prone areas, flow generation, assessing impacts of storage reservoirs on flood control, etc.	7: **Watershed and sediment management**: scope will be evaluation of impacts of alternative land use/cover on hydrology of the river system, estimation of sediment yield, and reservoir sedimentation.
4: **Coping with droughts**: priority shall be to support decisions in preparation of drought management plans, including planning for adaptation to climate change and variability.	8: **Navigation**: focus will be to identify how navigation might be affected by planned water resources development scenarios and support efforts to minimize the adverse impacts.

The riparian countries agreed upon eight thematic focus areas (see Table 9.2) for which NB DSS shall provide the necessary tools to support water resources planning and decision making. Climate change and water quality were identified as cross-cutting issues.

The key questions given in the beginning of this section were then addressed for each thematic focus area. Here, two thematic areas have been selected (see Table 9.3) to illustrate how a high level scoping of the DSS was made, guided by the key questions.

The thematic areas were used to develop a conceptual design of NB DSS. A list of DSS functional and non-functional requirements was then developed.

The second stage of user requirements elaboration and software specification followed once the conceptual design was finalized involving category two users. The second stage focused on detailed software requirements elaboration, design, development and deployment by involving a core team of professionals (future DSS users) from the Nile Basin countries and consultants. Based on their expected type of interaction with the DSS, these users were further grouped into user types and classes (Tables 9.4 and 9.5).

Table 9.3 Examples of DSS high level requirements

Thematic focus area 1: water resources development	
Sample decisions to be supported	*Examples on types of outputs from NB DSS for decision making*
1. Decision on selection of investment alternatives (e.g. small number of large, basin level water storage facilities, or large number of local level storage facilities, or a combination of these?) 2. Configuration of the overall (macro level) system-wide water development plan 3. Determination of features of development schemes (location, scale of development, size of components) 4. Identification of optimal operation rules (at the planning level)	Bio-physical/environment 1. Change in volume of water available: System wide (water balance, minimum flow); or at designated points in the river network (such as environmental hotspots, other points of interest) 2. Change in sediment movement downstream 3. Effects on navigable water reaches (draft, length of reaches, etc) Socio-economic 1. Financial and economic internal rate of return (FIRR/EIRR) (or B/C ratio) of alternative; 2. No of people to be re-located (from reservoir area)
Thematic focus area 2: optimal water resources utilization	
Sample decisions to be supported	*Examples on types of outputs from NB DSS for decision making*
1. Decision on selection of alternative ways of increasing system efficiency, e.g. 2. Basin wide resource priced demand management mechanism 3. Investments in water saving infrastructure and equipment 4. Subsidising of increased engineering efficiency at user level 5. Introduction of penalties for wastage 6. Decisions on changes to operating rules for dams to maximize benefits	Bio-physical/environment 1. Net gain in water availability at basin and specific locations 2. Impact (+/−) of contemplated alternative on downstream water flow Socio-economic 1. Net financial and economic productivity of water at basin level 2. FIRR/EIRR (or B/C) of alternative; or economic and financial unit costs of increased water

Table 9.4 User types

User type	Typical role in using DSS
Hydrologist	Preparation of hydro-meteorological and other data for modeling
GIS specialist	Geo-spatial data processing as input to modeling as well as presentation of outputs
Modeler	Setup/configure and execute models
Decision maker	End user of results of analysis
Water resource planner	Analyze model simulation output
System administrator	Exporting and importing study data, managing system users and studies
Water resources economist	Configure optimization problems
Software engineer	Coding plug-ins to integrate model tools with the NB DSS

Table 9.5 User class and roles

User class	Typical role in using DSS
Reviewer	A user that can read but cannot modify data; is allowed to make simple reports and analyses
Study reviewer	A reviewer who can read data for studies to which he/she is associated
Study member	A study reviewer who can change, delete and create data for studies to which he/she is associated
Study lead	A study member who can associate users as study reviewer or study member with the studies
Data owner	A reviewer who can add, change and delete data at the global level
Administrator	Perform backup, restore, data synchronization, add and delete users, set user access levels, create workspace

The user types from Table 9.4 can fall in any of the user classes given in Table 9.5. User class definition was used to define non-functional requirements of the DSS, such as user privileges in an organizational setup where the DSS is used by a number of users as a collaborative tool.

Use cases were used to elaborate and document user requirements. They present user interactions with the DSS and are used to identify, clarify, organize and document user requirements. Five use cases were selected that capture the range of possible DSS applications. These were broken down into steps to describe user actions and system responses. The steps were grouped into common steps [called Generalized Use Cases (GUC)] that are sufficiently detailed for use in requirements elaboration (Fig. 9.3).

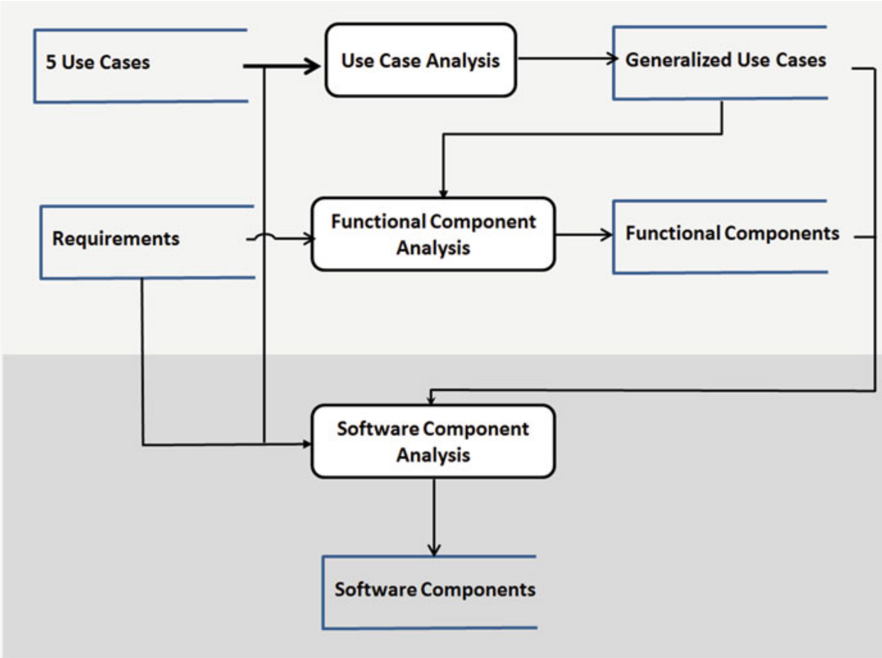


Fig. 9.3 From use cases to functional and software components

To illustrate how use cases supported the identification of detailed user requirements and DSS design, an example of a use case is provided in the form of a UML diagram (Figs. 9.4 and 9.5).

The UML diagram shown in Fig. 9.4 provides an overview of the workflow. Further breakdown of each element of the workflow was made into Generalized Use Cases (GUC) before DSS components were identified. Examples of GUC for the Use Case 04 are shown in Fig. 9.5. The detailed requirements and identification of features of each DSS component were then made in a series of requirements elaboration sessions by the core team together with consultants.

The second group of requirements was non-functional requirements. These requirements are used to describe the general features of the DSS that provide the overall environment for user experiences. Two examples of non-functional requirements are given in Table 9.6.

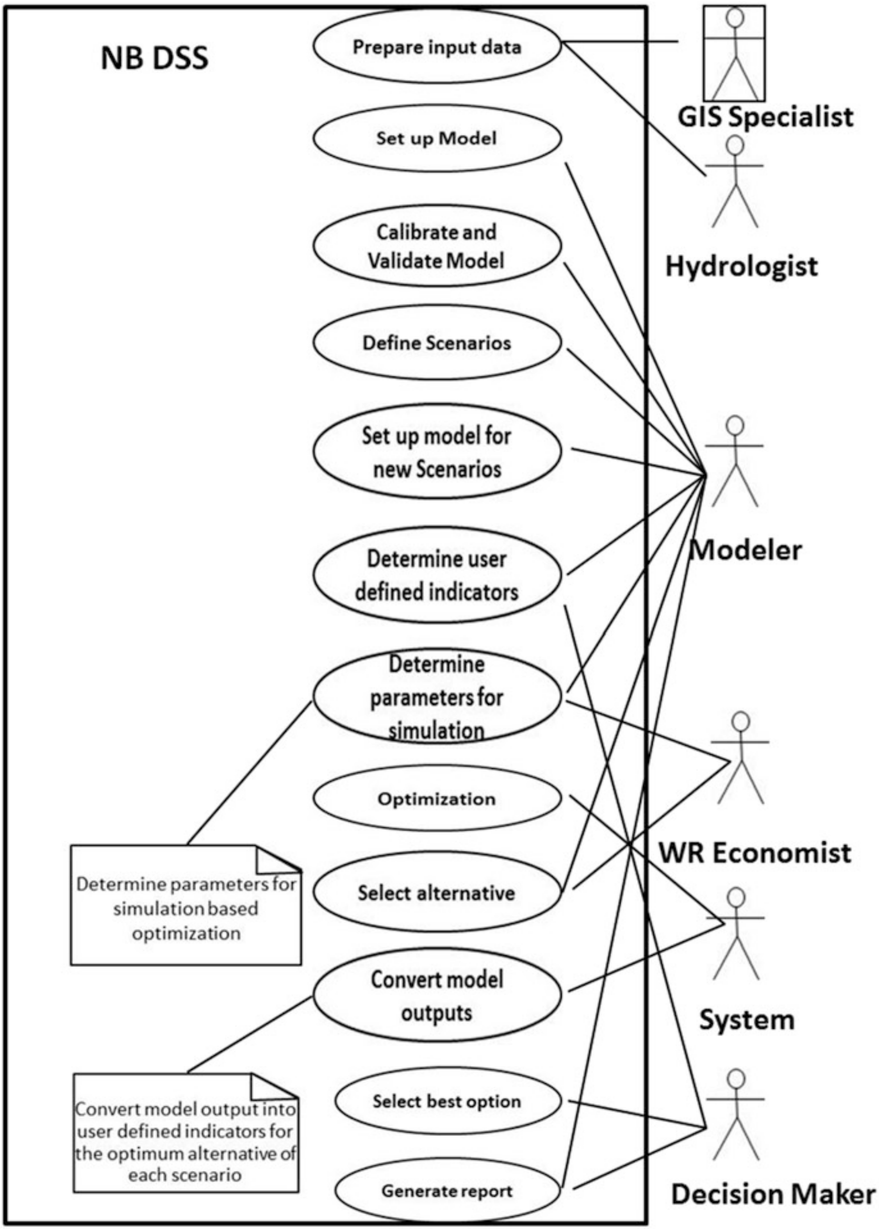


Fig. 9.4 Use case 04—overview UML

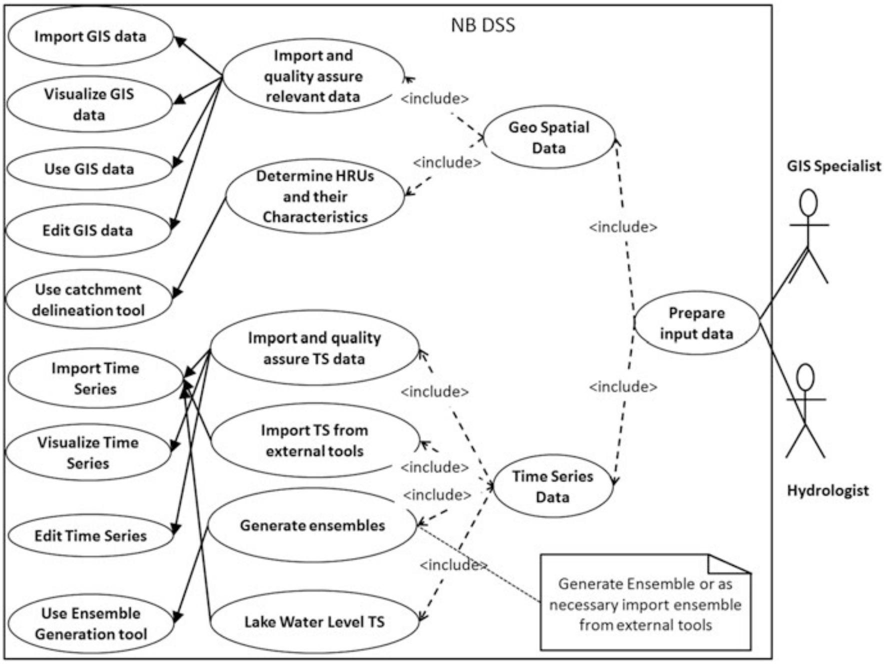


Fig. 9.5 Use case 04—detailed UML

Table 9.6 Examples of non-functional requirements

Examples of requirements	DSS features
Category: Information sharing and exchange	
Key functionalities: (a) Merge, append; (b) Synchronization of data between installations (in different countries and/or institutions); (c) Resolve data conflicts with user intervention	Data and information are stored within a 'workspace' with all data having its own system-generated 'change log' and user generated metadata. Exchange of data between installations is carried out by exchanging workspaces
Category: Audit trail	
The DSS should support tracking of changes in all user generated information (model setups, scenario configurations, time series data)	DSS keeps track of all changes to its stored data as part of its automatically generated 'change log'. The change log cannot be edited by users

9.4 System Design and Components

In order to meet the identified requirements NB DSS is designed to operate with a number of interlinked components (Fig. 9.6).

The DSS Front-end component is a Windows application providing all functionalities. It is built from "scratch" as part of DSS development.

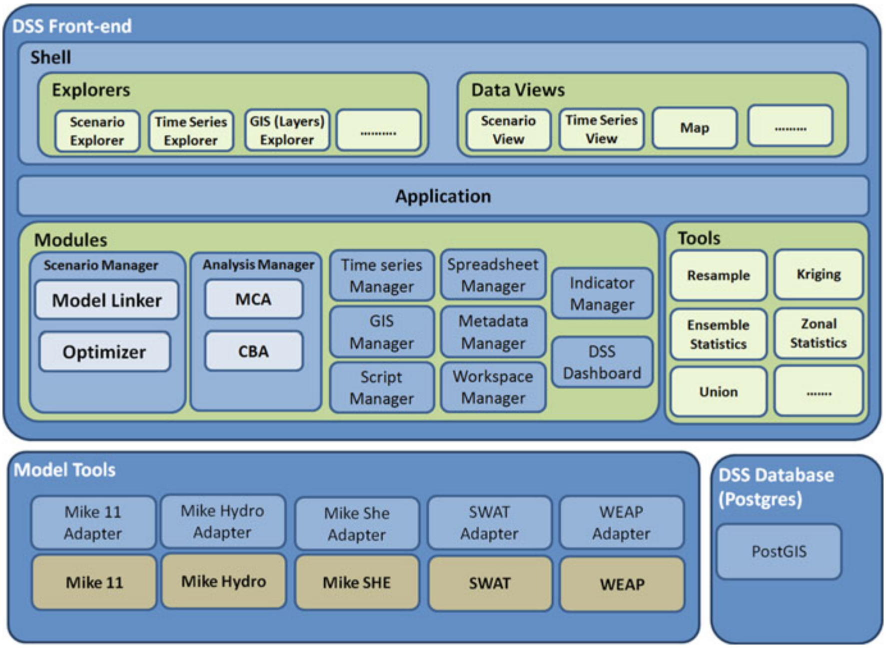


Fig. 9.6 NB DSS (logical) design layout with its components

The Database component is a Relational Database Management System (RDBMS) prepared for handling all types of DSS data—such as GIS (spatial) data, time series data, metadata, hydro objects (e.g. reservoirs, canals, water users, etc.) and scenario data.

The Model Tools component is a collection of generalized mathematical models (proprietary or public-domain). These are "off-the-shelf" products and as such are not directly part of the DSS Front-end.

The NB DSS software framework is organized into functional components that provide specialized toolsets. Major building blocks are termed as 'Managers' with task-specific functionalities under each Manager. In addition, a few tools provide generic, i.e. not Manager specific, functionality, such as data import/export, and metadata schema import.

Simulation models are key components of NB DSS. Given that Danish Hydraulic Institute (DHI) has been main consultant for NB DSS development the initial set of modeling tools are from the MIKE family of DHI. MIKE BASIN (recently changed to MIKE HYDRO BASIN) is the main river basin simulation model for analyzing water budgets and allocations. It can be combined with MIKE 11- one-dimensional river hydraulic model for detailed hydraulic analysis, MIKE SHE—detailed hydrological model for analyzing catchment hydrology, or NAM (simpler hydrological-rainfall runoff model—not depicted in Fig. 9.6). For analyzing different alternatives these models need to be set up separately, outside of

NB DSS, using their own user interfaces, and then be ‘registered’ in NB DSS as different model setups. For this purpose NB DSS has corresponding ‘adapters’ as depicted in the bottom part of Fig. 9.6. These adapters have two main functions: configuring and storing the model setup in the database and performing simulation runs. The system is designed to be extensible for use of other model tools, for which adapters need to be developed. The current version of NB DSS already has adapters for two additional models: WEAP [24] and SWAT [27] (another semi-distributed hydrological model). After registering, model setups become available in NB DSS for subsequent analysis, by other components.

Many of the decisions in the eight thematic focus areas of the DSS (samples given in Table 9.2) involve evaluation of alternatives and selection of most acceptable alternative. In NB DSS vocabulary, scenarios are alternatives that need to be evaluated, possibly combined with varying external conditions (e.g. climate change or socio-economic development).These alternatives could involve different configurations of water resources infrastructure, strategies for allocation of water to various users, strategies for operation of dams, or varying scales of developments of the water resources for a specific purpose, say, irrigated agriculture. Thus, an important step in the decision making process is the creation of a set of alternatives (scenarios) that shall be evaluated against agreed upon criteria.

In NB DSS, formulation of these alternatives is carried out through the use of the Scenario Manager. The scenario manager uses registered model setups, which represent the underlying basic configuration of the water resources system of the study under consideration, and helps the user to create variants of the system, i.e. alternatives. These variants represent alternative planning or management scenarios created by varying model input data or parameters. The scenario manager is used to run simulation of scenarios by calling the model engine, and save all simulation outputs in the DSS database. Once simulation runs of the various scenarios have been completed, the next step in the decision making process starts, that is evaluation of criteria and running the MCA session.

In NB DSS, alternatives (scenarios) are evaluated against (performance) criteria that reflect interests of stakeholders and decision makers. These criteria are generated based on a set of indicators computed in the DSS from the outputs of simulation runs. A criterion is defined based on a set of indicators. As an example, the evaporation losses from a particular dam can be an indicator computed based on simulation output. A criterion that uses this indictor could be the sum total of evaporation from all dams in a study area. This criterion can be used to assess overall (system-wide) water loss. Therefore, computation of indicators is an important first step in the definition of evaluation criteria.

In NB DSS, indicators are computed using user-defined algorithms that relate simulation outputs to the specific indicator in question. These algorithms are then converted into indicator scripts, which are pieces of program code that are run by the DSS to compute indicator values; DSS uses Iron Python as scripting language. In any DSS application, users can either generate their own scripts or use any of the available scripts in the DSS database. Once the indicators are computed, criteria are defined based on available indicators, which are then used to evaluate the alternatives.

The MCA tool offers users the features for defining criteria (based on available indicators), select a set of scenarios for evaluation, define user preferences in terms of weights for each criterion and synthesize results of evaluation in a 'Decision Matrix'. To reflect stakeholder preferences for specific criteria, weights can be assigned to the various criteria. Differing weighting schemes can be employed on the same set of scenarios and criteria. This way, effects of weights on final ranking of scenarios can be analyzed. A dedicated tradeoff analysis toolkit can be used to compute tradeoffs among a few selected scenarios that have been picked from the rankings of scenarios made as per different weights of criteria by various stakeholder groups.

Different models (e.g. MIKE BASIN and MIKE 11) can be linked together, or model setups for different spatial parts of the basin can be linked together, which are tasks of the Model linker component of the Scenario Manager. The Optimizer component enables usage of simulation models in optimization frameworks (single or multi-objective), using one of the search algorithms provided. Currently the DSS provides five search algorithms: the Non-Dominated Sorting Genetic Algorithm II (NSGA-II), Shuffled Complex Evolution (SCE) with its simplex variant, the Dynamically Dimensioned Search (DDS), and Monte Carlo.

Scenarios can be evaluated from within the scenario manager by simple comparison. More comprehensive evaluation/analyses however, are provided by the MCA and CB tools of the Analysis manager, where scenarios are evaluated against multiple objectives (using the defined indicators) and standard economic models. These tools need to be used by groups of stakeholders during dedicated workshops to generate consensus about certain aspects of the evaluation; the MCA tool has a feature for varying weights attached to criteria to take into account stakeholder preferences.

The remaining 'managers' support various generic tasks, such as data processing, preparing scenario evaluations, further analysis of results, reporting and overall system administration. Time series manager and GIS manager are used for managing temporal and spatial data, respectively.

A Spreadsheet Manager is provided for performing tabular calculations similar to Excel. The Metadata manager uses catalogues of metadata for all data available in the system. Finally, managers for system administration, reporting and management of case studies are also provided.

9.5 System Implementation

The DSS development process followed a number of steps from detailed design to development, testing and application (Fig. 9.7).

In the inception phase, the DSS software requirements were elaborated and the software architecture design was developed. The development process involved three stages (cycles) where a fully functional DSS software is delivered at the end of each cycle, which would then be delivered as 'DSS Release x'. This release was

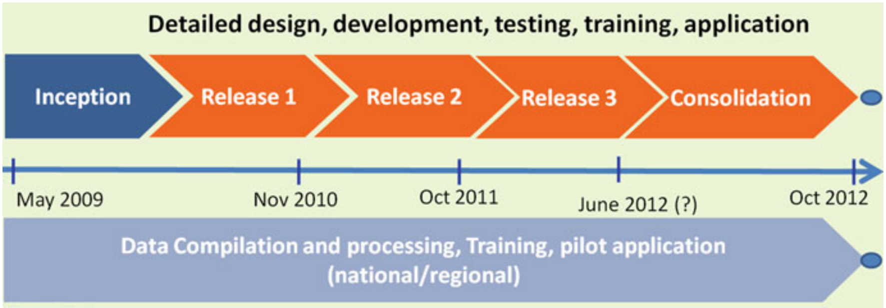


Fig. 9.7 NB DSS development process

then subjected to User Acceptance Test. Parallel to the DSS development cycles, especially once the first release was delivered, independent testing of the software and pilot application of the same were carried out by the core team. Throughout the DSS development phases, user trainings were conducted fairly extensively.

The institutional setup comprised of a regional DSS center in Addis Ababa, Ethiopia and national DSS units—one in each NBI member country. A core team of about 40 professionals drawn from the Nile Basin countries was formed to spearhead the DSS design, development, testing and final deployment. These professionals were distributed in the regional DSS center, national DSS units and national ministries of water affairs. In addition, a regional DSS network and nine national DSS networks—one in each country—were formed.

The core team was responsible for driving the requirements analysis (including use case analysis), providing technical inputs to the design process, review and approval of design, DSS software testing and review software testing results carried out by the developers (DSS contractor).

Once the conceptual design was prepared, the NBI contracted two consulting firms and one individual consultant for the development, pilot application and independent testing of the software releases, respectively. Three international advisors made up a panel of experts that provided technical advice at strategic level while other consultants were hired as needed on operational level.

DHI (Denmark) was hired as main DSS contractor while Aurecon (South Africa) led the pilot application of the DSS. An individual software tester assisted NBI in designing and carrying out software testing.

The DSS software was developed based on the .Net framework. PostgresSQL is the database management system with the PostGIS extension for handling spatial data. The DSS client is a Windows application while the database can be deployed either on a Windows or a Linux machine. The DSS software doesn't require a third-party software license from the user in order to be run.

9.6 User Interface Design

The GUI of the DSS was designed based on detailed user requirement analysis. The requirement specifications and design (supported by Mockups) were carried out in a number of working sessions involving the consultant (DSS contractor), the core team and, as required, advisors.

One of the challenges in the design of the UI was how to balance user requirements for flexibility with ease of use of the system. In addition, the degree of coupling of modeling tools into DSS was another design decision that also reflected on the UI design. The modeling tools remain external to the DSS, hence they retain their own UI and, consequently, the design paradigm of the DSS UI didn't apply to those of the modeling tools. This means that the user needs to get used to different UIs when using the complete DSS.

As a result, the DSS GUI evolved as an IDE-style user interface with standard controls that are consistent across all modules. Controls are either Explorer Controls or Data View Controls to maintain a transparent and easily recognizable UI structure. Explorer Controls provide access to the module functionality.

To avoid lengthy context menus and complex pop-up dialogs, and to maintain consistent structure across Modules and Controls, the DSS Shell offers a set of standard controls that interface with the module controls. These include a Property Control and a Toolbox Control. When modules are loaded into the system, the controls are automatically grouped and displayed by type. This is illustrated in Fig. 9.8.

Fig. 9.8 User interface of NB DSS with the different types of controls

The task of the Property Control is to expose the properties of the “object(s)” that are currently selected in the active control. “Objects” can be any “selectable” UI element residing in either an Explorer or Data View control, e.g. a line in a chart, a GIS layer, or a column in a table. The Toolbox Control displays a list of tools that can be applied to the currently selected object(s). This means that if a time series is selected, the toolbox shall list the available time series tools. When a tool is selected it shall be configured in the Property Control.

The user interface of the DSS was designed with a fairly high degree of flexibility. A user can have access to all the data in the workspace he/she is logged into. Regardless of which tool the user invokes in the main DSS system (not in the modeling tools), the user experiences the same UI, which is a ‘plus’ for the UI design. However, every user will use the same user interface not customized to the type of work he/she is interested to accomplish using the DSS. For advanced users, this flexibility is a plus. Experience shows that the DSS UI could simply be overwhelming for some users. Those users that just need to accomplish a very specific and simple task often find it too much to use the full DSS UI.

9.7 Cases Analyzed with NB DSS

This section introduces characteristic cases analyzed by employing the NB DSS. These covered regional (multi-country) cases as well as cases concerned with national water resources issues. Two regional (basin, sub-basin wide) and one national application cases are presented below.

The first sub-basin level application was in a multi-sector investment planning study in the Nile Equatorial Lakes region. The DSS, especially the modeling system, was used to analyze a range of water resources development and management scenarios that address projected future water demand for the planning horizon of 2035. The DSS River System Model was coupled with economic assessment tools (custom-built for the analysis) to evaluate and select—on the basis of riparian country-agreed criteria—a preferred scenario. The selected scenario was taken further to work out the multi-country investment plan for the region.

The NBI Secretariat (Nile-SEC) is employing NB DSS to make water demand and supply projections in the 2050 time horizon and thus identify potential constraints. The objective of this exercise is to identify strategies for addressing potential shortfalls to meet the water resources demands of development projects planned by riparian countries. The DSS is being used for modeling, creation of scenarios, evaluation of scenarios and synthesizing analysis results. Selected indicators were used to describe the performance of the Nile Basin in meeting the current and future water demands. Examples of these indicators are:

- Reliability of water supply (by demand center)
- Energy reliability, energy production
- Water losses through seepage and evaporation (locally or system wide)

- Unmet water demands
- Overall basin yield
- Total available water storage
- River flow at selected notes (regulation indicator)
- Deviation from historical flow patterns (at a point)
- Reduction and changes in timing of flood peaks

The first step was to establish the baseline Nile Basin model in the DSS. The baseline describes water demand and use for key sectors (Municipal and Industrial, Irrigation, Hydropower production) as of 2014. The baseline model was availed to each riparian country as DSS workspace through their national DSS units for any further review and use in their own analyses. The DSS workspace packages all data, models, and analysis results and is easily importable into national DSS installations. This way, the DSS is serving as a common platform for information sharing, analysis and creating common understanding about the Basin's water resources.

Once the agreed baseline was established, riparian countries availed data on their planned water resources development projects (e.g. expanding irrigated agriculture; hydropower) to Nile-Sec. The expert group thus made use of the assembled data to generate possible future scenarios of growth in water demand, taking the entire Nile basin as a planning unit. These scenarios were used to answer the following questions regarding the future of Nile Basin water resources:

- Will there be sufficient water to meet water demands of all planned projects?
- How big will the shortfall be?
- Which parts of the Nile Basin will likely experience biggest shortfall?
- How much will improvements in irrigation efficiency reduce the potential shortfalls?
- What will the effect of climate change be on exacerbating the magnitude, frequency of the shortfalls?

So far, the analysis has shown the state of current (2014) balance between water demand and supply in each riparian country; projected water demand based on plans by riparian countries and expected future shortfalls. The analysis has demonstrated that although improvement in irrigation efficiency can contribute to partially reduce potential shortfalls, it is not sufficient to address the entire shortfall in the system. The results of the analysis are now used to design specific measures for addressing the expected shortfalls and enhancing the efficiency of water use in the basin.

In Uganda, Nile Basin DSS has been used to formulate the national water resources management strategy. The strategy is intended to provide a framework for management and development of the country's water and related resources up to 2040. The DSS was used to assess the hydrological implications of key national water resources development objectives; the possible trade-offs between WR development options with focus on HP generation, wetland and upland irrigation and downstream flow; and to compare the pros and cons of the options, including environmental flow.

Recently, the DSS was also used to estimate the hydrological implications of the new Grand Ethiopian Renaissance Dam (GERD) and other planned dams in Ethiopia on Sudan. The study was made by Sudanese team and focused on impacts of the dams on the Blue Nile system in the Sudan. It studied the impacts from changes in flow regimes (high and low flows), reservoir water levels, and impacts of hydropower generation.

One other national application of NB DSS is for preparing the integrated water resources management plan of the Lake Tana Sub-basin of Ethiopia. The DSS was used to quantify and evaluate the impact of irrigation developments of Tana Sub Basin on: Tana Beles Hydropower Production; Lake Tana Navigation; Lake Tana Fish Production and System Evaporation. This study was carried out by the Abbay Basin Authority of Ethiopia.

9.8 Experiences and Future Prospects for NB DSS

Since first version of NB DSS was released, through a number of case studies and user forums (DSS user community), NBI has been able to interact with users and get insight into their experiences.

One of the main positive experiences of using NB DSS is that it provided a common analytic platform for water resources analysis and exchange of technical information. It has helped in putting in place a common workflow and toolsets across the Nile riparian countries. Users can easily share data across countries by exchanging DSS workspaces.

The DSS provides the required analytic foundation for collaborative decision making and can greatly enhance the deliberations on cooperative water resources management of transboundary rivers. However, experience shows that such analytic system can only make a difference within a context of a (political) decision making process. Decision making in multi-country water resources management is essentially a political process in which technical information plays a critical role. In the context of the Nile Basin, due to the inherent political complexity of the decision making process, the DSS has so far made a very modest contribution in influencing major decisions. However, it is gradually gaining the necessary attention by political leaders as more and more of its uses are generating information that decision makers found very useful in their understanding of the Nile Basin water resources.

The integration of data processing, modeling, scenario management and multi-criteria analysis tools in one system was found to be positive experience by users. The DSS is packed with a diverse set of tools. Those users that are good in writing scripts, found the Script Manager very helpful for automating their work, for instance, by batch processing of tasks. However, this versatility of the DSS has its costs. Many users use a small proportion of the tools. Some of DSS tools are used very infrequently because they are not required in day to day water resources planning activities. One may ask whether it would have been more appropriate for the DSS to have a narrower scope than it has right now. For example, the

DSS could have been packaged in three flavors, data management, modeling and scenario analysis; and decision making. The concept of indicators implemented in the DSS is found innovative by its users. The DSS provides the flexibility for users to implement their own methods for estimating indicators. However, a major challenge of this is lack of data that support the causal link between many indicators (such as, say, a habitat suitability index for a specific aquatic animal species and the hydraulic parameters of a river or a lake). The fact that users can develop their own scripts for computing indicators based on model simulation outputs provides the flexibility to continuously improve the indicator definitions. However, lack of data still hampers use of the full capacity of the indicator tool.

The DSS is a desktop application, which limits the number of users that can access it. The DSS was not designed as a web-based application. This was a conscious design decision made factoring in the weak internet connectivity in many of the Nile riparian countries at the time of the DSS development. This is changing and NBI is considering ways for providing web access to data and information in the DSS.

The original version of the DSS uses the Mike family of modeling tools (by DHI). Since these modeling tools are proprietary with fairly high license costs, it wasn't possible to issue large number of licenses to potential users. This was one of the feedbacks from potential users. The NBI has added new adapters for WEAP and SWAT to enable the DSS use with these two modeling tools in addition to the Mike family of modeling tools. These new additions come with 300 more licenses (for the DSS proper) and are expected to expand the user circle. Simplifying the User Interface (UI) is one of the priority future improvements the NBI is planning for the DSS. To cater for the needs of the expanding circle of users, NBI is developing online training courses that will be available free for interested users.

9.9 Conclusions

From the NB DSS design, development and its initial use it can be concluded that this is a tool with a promising potential for analyzing water resources problems in the complex Nile basin and for supporting both national and trans-boundary decisions regarding water resources. Key condition for successful NB DSS development has been the process of user requirements gathering and analysis, which has been realized through active engagement of all stakeholders from the riparian states. Although challenging and time consuming this condition is necessary for developing the sense of ownership and acceptance of this DSS.

The availability of NB DSS however, is just one step towards developing comprehensive, science-based water resources solutions in the Nile basin that would offer shared benefits for the different countries. The political will for broader cooperation among the riparian states provides the necessary condition under which the DSS can be applied and provide the support that was initially envisaged.

NB DSS also needs to become more widespread, while keeping its flexibility and adaptability to new demands from the stakeholders and to new technologies. For this purpose a sustained and long term capacity development strategy is being implemented, targeting both current NB DSS users and future water resources experts.

References

1. Almoradie, A., Cortes, V., Jonoski, A.: Web-based stakeholder collaboration in flood risk management. J. Flood Risk Manag. **8**(1), 19–38 (2015)
2. Andreu, J., Capilla, J., Sanchís, E.: Aquatool, a generalized decision-support system for water-resources planning and operational management. J. Hydrol. **177**(3), 269–291 (1996)
3. CSU: MODSIM-DSS. Colorado State University. http://modsim.engr.colostate.edu/index.shtml (2016). Accessed 20 May 2016
4. de Kok, J.L., Wind, H.G.: Design and application of decision-support systems for integrated water management: lessons to be learnt. Phys. Chem. Earth Parts A/B/C **28**(14), 571–578 (2003)
5. Delipetrev, B., Jonoski, A., Solomatine, D.P.: Development of a web application for water resources based on open source software. Comput. Geosci. **62**, 35–42 (2014)
6. Deltares: RIBASIM (River Basin Simulation Model). https://www.deltares.nl/en/software/ribasim/ (2016). Accessed 20 May 2016
7. DHI: MIKE HYDRO BASIN River basin management and planning. https://www.mikepoweredbydhi.com/products/mike-hydro-basin (2016). Accessed 20 May 2016
8. Georgakakos, A.: Decision support systems for water resources management: Nile basin applications and further needs. In: CPWF: Proceedings of the Working Conference, pp. 23–26 (2006)
9. Giupponi, C., Sgobbi, A.: Decision support systems for water resources management in developing countries: learning from experiences in Africa. Water **5**(2), 798–818 (2013)
10. GWP: The role of decision support systems and models in integrated river basin management. Global Water Partnership (2013)
11. Hämäläinen, R., Kettunen, E., Marttunen, M., Ehtamo, H.: Evaluating a framework for multi-stakeholder decision support in water resources management. Group Decis. Negot. **10**(4), 331–353 (2001)
12. Jamieson, D., Fedra, K.: The waterware decision-support system for river-basin planning. 1. Conceptual design. J. Hydrol. **177**(3), 163–175 (1996)
13. Jonoski, A.: Hydroinformatics as Sociotechnology: Promoting Individual Stakeholder Participation by Using Network Distributed Decision Support Systems. Sweets & Zeitlinger B.V., Lisse (2002)
14. Jonoski, A., Evers, M.: Sociotechnical framework for participatory flood risk management via collaborative modeling. Int. J. Inf. Syst. Soc. Chang. **4**(2), 1–16 (2013)
15. Liu, Y., Gupta, H., Springer, E., Wagener, T.: Linking science with environmental decision making: experiences from an integrated modeling approach to supporting sustainable water resources management. Environ. Model. Softw. **23**(7), 846–858 (2008)
16. Loucks, D., Van Beek, E.: Water Resources Systems Planning and Management: An Introduction to Methods, Models and Applications. UNESCO Publishing (2005)
17. Matthies, M., Giupponi, C., Ostendorf, B.: Environmental decision support systems: current issues, methods and tools. Environ. Model. Softw. **22**(2), 123–127 (2007)
18. Mysiak, J., Giupponi, C., Rosato, P.: Towards the development of a decision support system for water resource management. Environ. Model. Softw. **20**(2), 203–214 (2005)

19. NBI: State of the River Nile Basin Report: Nile Basin Initiative. http://nileis.nilebasin.org/content/state-river-nile-basin-report (2012). Accessed 20 May 2016
20. NBI: Nile Basin Decision Support System: Nile Basin Initiative. http://nbdss.nilebasin.org/support/home (2016). Accessed 20 May 2016
21. Pahl-Wostl, C., Craps, M., Dewulf, A., Mostert, E., Tabara, D., Taillieu, T.: Social learning and water resources management. Ecol. Soc. **12**(2), 5 (2007)
22. Pierleoni, A., Camici, S., Brocca, L., Moramarco, T., Casadei, S.: Climate change and decision support systems for water resource management. Proc. Eng. **70**, 1324–1333 (2014)
23. Power, D.: Computerized decision support case study research: concepts and suggestions. In: Papathanasiou, J., Ploskas, N., Linden, I. (eds.) Real-World Decision Support Systems. Springer, Heidelberg (2016)
24. SEI: WEAP Water Evaluation and Planning. http://www.weap21.org/index.asp?NewLang=EN (2016). Accessed 20 May 2016
25. Serrat-Capdevila, A., Valdes, J.B., Gupta, H.V.: Decision support systems in water resources planning and management: stakeholder participation and the sustainable path to science-based decision making. In: Efficient Decision Support Systems: Practice and Challenges–From Current to Future. Open Access Publisher (2011)
26. Sulis, A., Sechi, G.M.: Comparison of generic simulation models for water resource systems. Environ. Model. Softw. **40**, 214–225 (2013)
27. SWAT: Soil and Water Assessment tool SWAT. http://swatmodel.tamu.edu (2016). Accessed 20 May 2016
28. Walsh, M.R.: Toward spatial decision support systems in water resources. J. Water Resour. Plann. Manag. **119**(2), 158–169 (1993)

Chapter 10
The AFM-ToolBox to Support Adaptive Forest Management Under Climate Change

Manfred J. Lexer and Harald Vacik

Abstract Climate change may strongly impact on forests and affect the provisioning of forest ecosystem services. The identification, design, selection and implementation of adaptive measures in forest management requires a sound knowledge base as well as tools to support the forest manager in decision making. Decision support systems (DSS) are considered as particularly useful to assist in dealing with ill structured decision making problems. Forest management planning and decision making deals with highly complex socio-ecological systems with multiple interacting spatial and temporal dimensions. Finding ways and means to communicate findings about such complex relationships in forest ecosystems and their management via information technology is a challenge in itself. This is amplified if decision problems include land use change and climate change issues as uncertainty in planning outcomes increases. The literature reports numerous attempts to develop DSS for forest management. However, recently several review papers conclude that there has been only limited uptake of DSS into practice because frequently user demands and the characteristics of decision problems are not considered properly. In this contribution we propose five design principles for forest management DSS: (1) modularity, (2) accessibility via the internet, (3) inclusion of different types of knowledge and information, (4) possibility to use different data sources, and (5) support of specific problem types. Based on these principles we promote a ToolBox approach attempting to meet context specificity as well as flexibility addressing different user and problem types simultaneously. The AFM (Adaptive Forest Management) ToolBox is introduced and the conceptual design and technical implementation of the ToolBox is presented. The combination of different decision support techniques (e.g. vulnerability assessment, multi-criteria analysis, optimization) allows to support all phases of the decision making process and provides the user with the flexibility to interpret the information in various forms. The results of a self-assessment of the ToolBox against eight evaluation criteria for DSS are combined with a feedback from a panel of expert users who

M.J. Lexer (✉) • H. Vacik
University of Natural Resources and Life Sciences, Vienna Peter-Jordan-Strasse 82, A-1190 Wien, Austria
e-mail: mj.lexer@boku.ac.at; harald.vacik@boku.ac.at

© Springer International Publishing Switzerland 2016
J. Papathanasiou et al. (eds.), *Real-World Decision Support Systems*,
Integrated Series in Information Systems 37,
DOI 10.1007/978-3-319-43916-7_10

had tested the usability and had evaluated the conceptual approach of the ToolBox. The feedback stimulates the further development and ultimately will increase the level of acceptance by potential users. It is concluded, that the ToolBox approach focusing on modularity while avoiding over-emphasis of technical integration provides the right frame to secure the flexibility to add new tools and improve the support of decision making processes which is mandatory if a DSS should be taken up by practice.

10.1 Introduction

Forest management paradigms have undergone drastic changes over the last decades, and emerging concepts such as ecosystem management and sustainable forest management (SFM) are increasingly adopted as framework for decision making in forest management (e.g., [3, 13, 26]). These concepts inter alia advocate a science-based decision making process and a continuous evaluation and adaptation of management in the light of emerging scientific understanding and shifting societal objectives.

Climate change and its impacts on forests and the provisioning of forest ecosystem services has become one of the key issues in forest science as well as in practical forest management (e.g. [14, 17]). Forests, due to their longevity, are expected to be strongly affected by the rapid rate of climate change projected for the twenty-first century [2], straining the potential of natural adaptation processes. Challenges for forest resource managers will include the need to balance the societally required portfolio of services and functions [16, 35]. Numerous studies about climate change impact and vulnerability assessments as well as on adaptive and "climate-smart management" (e.g. [18]) at various spatial scales, from local to continental, exist in the literature (compare [17]). However, a consistent integration in strategic and tactical planning and decision making processes in forest resource management is still widely missing [11, 27]. A forest manager, facing the challenge to decide about forest management strategies and operational plans, can screen the available literature to search for representative case studies, or hire a consultant to conduct a study on her/his own (see [6]). In the former case, approaches in the literature are extremely heterogeneous and comparability of results is difficult for a non-expert. In the latter case, planning and outcomes may benefit from a harmonized guideline to adaptive management planning as well.

Decision Support Systems (DSS) have proven to be suitable platforms for the integration of information, models and methods required to support complex, ill-structured and value laden forest management problems (e.g. [5, 22, 23, 30]).

Also decision making situations that involve many stakeholders and natural resources need to rely on tools that support and facilitate the explicit inclusion of stakeholder preferences in the decision making process [10]. DSS have the potential to play an important role in facilitating participatory planning processes [12, 15, 31]. A IUFRO Task Force [23] underlined the impacts of decision models and systems on both the efficiency and the effectiveness of forest management. The European Union (EU) Forest-Based Sector Technology Platform further emphasized the importance of advanced and innovative planning methods and computer tools for addressing multifunctionality and sustainability of forest management as a key issue for the competitiveness of the forest sector. The latest compilation of forest management decision support systems is found in the report [5] and special issues [4, 33] of the related European Cooperation in Science and Technology (COST) Action FORSYS.

Examples for DSS in forest management include, for instance, NED-2 [29] in the United States and HEUREKA (http://www.slu.se/SHa) in Sweden. However, transferability to regions other than those for which DSS originally had been developed for is limited and requires huge efforts in adapting the knowledge base and in model development and calibration. Moreover, neither NED-2 nor HEUREKA use predictive forest simulation models which can capture the effects of a changing climate. Currently there exists no forest DSS addressing the specific needs of adaptive management under climate change.

While the potential usefulness of dedicated DSS for adaptive forest management is obvious, at the same time attempts to develop forest related DSS doesn't fully seem to meet the expectations of end users. This may origin in expectations being too high and not realistic. There may also be a lack of understanding by scientists what the real problems and decision support needs of end users are. To improve the effectiveness of DSS development a close interaction between developers and end users thus seems to be the most promising approach. Ultimately, this would imply that DSS development takes place within defined and specific contexts regarding stakeholder expectations and needs. For supranational R&D projects this in turn would imply intensive stakeholder interaction processes which are rarely foreseen due to reasons such as, for instance, budget constraints. Moreover, it is questionable whether it is possible at all to combine various specific stakeholder backgrounds and related DSS needs within one single development process.

In responding to this situation we set out to develop a generic DSS "toolbox" framework which can then be customized to specific contexts by further targeted follow-up R&D activities. At best, such a framework contains a combination of tools which the end user could combine according to the specific context. In this manner expert systems, model and data driven DSS can be linked in a web based system that allows easy access and supports the consideration of the specific needs of adaptive management under climate change.

In this contribution we introduce the design principles for the development of forest management DSS based on the analysis of the system requirements and promote the AFM (Adaptive Forest Management) ToolBox approach as a way to meet context specificity as well as flexibility addressing different user and problem types simultaneously. In the System Design section the conceptual design and technical implementation of the ToolBox is presented focusing on the main components ToolBox DataBase, the Vulnerability Assessment Tool, the Optimized Management Plan and a content management system. The section System Development provides an overview of the main steps in development of the AFM ToolBox. The different decision support techniques (e.g. vulnerability assessment, multi-criteria analysis, optimization) are presented in the section User Interface Design to demonstrate the way of how all phases of the decision making process are supported. The section System Implementation demonstrates how numerous open source technologies are employed in the AFM ToolBox. The self-assessment of the ToolBox against eight evaluation criteria for DSS combined with a feedback from a panel of expert users is presented in the section System User Experience.

10.2 System Requirements Analysis

In assessing possible reasons for slow and hesitant adoption of DSS in practice a few major general challenges can be identified. A crucial issue for the acceptance of DSS by users is whether the DSS targets the right user(s). A typical setting in forest resource management combines one responsible decision maker and a heterogeneous group of stakeholders having a diversity of partly contrasting interests and expectations towards forest management. Another important prerequisite for DSS adoption is that the problem type featured by the DSS is actually reflecting the decision making problem of a user [30]. Characterizing the problem type includes external drivers such as landuse and socio-economic changes and global phenomena such as climate change. The specific combination and relevance of these components may vary strongly among potential application cases. The decision maker might be interested to find the most appropriate management alternative for the provisioning of the desired ecosystem services (e.g. timber production, carbon storage, maintenance of biodiversity) considering future forest development including disturbances and changing environmental conditions. In order to select the best forest management the past, current and potential future alternative forest management regimes under a range of climate scenarios should be explored to make a rational choice. Forest types, spatial scales, disturbance regimes and demanded ecosystem services may vary largely from case to case. For the development of DSS this broad range of variability in problem type characteristics is particularly challenging in the context of forest management DSS.

If the intended user community is heterogeneous with regard to institutional background, role in decision making processes, available expert knowledge and interests regarding forest products and services it is likely that a DSS featuring a highly pre-determined decision making process (i.e. the decision model) will not be accepted. Moreover, beyond the different procedural approaches to decision making, it is obvious that a single decision support tool will not be sufficient to cover all needs of all decision makers and stakeholders. However, the consideration that context specificity and flexibility are key requirements for acceptance of decision support tools by end users calls for a tool box approach in which a diverse set of tools is made available to potential users and in which the mode of using these tools can be adjusted according to various decision making processes [1, 21].

Recent technological advances provide new options for the implementation of DSS and technical integration of DSS components is feasible, although high resource input is mandatory. However, emphasizing technical integration of DSS components may lead to decreasing flexibility with regard to decision making processes that a DSS can support and with regard to tools included in a DSS. Furthermore, seen from a larger perspective, a tool box approach provides the opportunity for continuing development work over several project life cycles. This may help avoiding the need to start DSS development multiple times from scratch.

Based on experiences in previous DSS development projects [15, 21, 31, 32] and on DSS literature five key requirements for the development of the AFM ToolBox have been identified:

Support Modularity The metaphor of a "tool box" points already at modularity: it should be easy to add new tools (also from third parties) or to exchange existing tools. Similarly, tools should be able to share common elements (e.g., administrative functionalities such as user management, data import and export, saving DSS sessions, printing).

Support Accessibility via the Internet Current technological advances allow the development of web-based decision support tools as internet browsers can run complex web applications which can be accessed due to the widespread availability of broadband internet connections. Specific advantages of a web-based approach are the reduced access barrier (no downloads and installations required) and the usability of decision support tools in a collaborative decision making situation.

Support Different Types of Knowledge and Information The ToolBox should support both interactive, data driven tools and "softer" types of information such as demonstration examples, documents, maps and FAQs.

Support the Use of Different Data Sources The AFM ToolBox should offer easy try-out of tools with ready-to-use data from case studies of different projects. It should also be easy to operate the tools with data from the own problem domain and region.

Target Different User- and Problem Types The ToolBox users are likely as diverse as the problems that the stakeholders are facing, and they have different levels of expertise. The ToolBox should therefore be able to serve managers, analysts, and scientists and support the analysis of specific forest management problem types.

10.3 System Design

For the general system design we have followed the five principles listed above: (1) modularity, (2) accessibility via the internet, (3) inclusion of different types of knowledge and information, (4) possibility to use different data sources, and (5) support of specific users and forest management problem types.

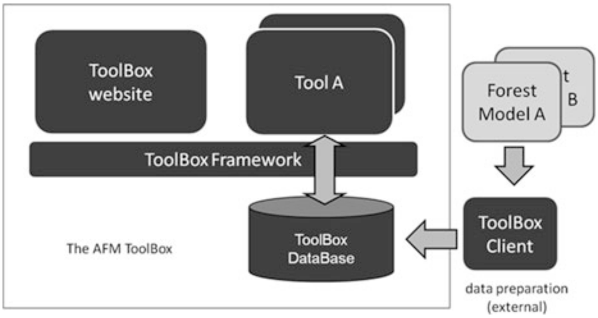


Fig. 10.1 System design of the AFM ToolBox. *Arrows* indicate the flow of data

The AFM ToolBox is based on a set of software applications aiming to support adaptive forest management under climate change. The main components of the ToolBox are the ToolBox DataBase, the Vulnerability Assessment Tool, the Optimized Management Plan, a content management system (CMS) providing access to different types of information and a collection of web-based tools which support the handling of data provided by forest simulation models. In the context of the DSS categorisation provided by Power (this book), the approach can be classified as Data-driven DSS. The input data for the tools in the ToolBox are all stored in the ToolBox DataBase. The CMS makes the "knowledge base" available to the end-user by means of static and dynamic web pages. The CMS is an integral part of the ToolBox, but not directly connected to DataBase and Tools. The ToolBox client serves as interface between external models and the DataBase of the AFM ToolBox. Figure 10.1 indicates the processes and flow of data needed to run the tools with data from the Database by the use of the CMS.

10.3.1 ToolBox DataBase and ToolBox Client

The Database supports the processing of data in a harmonized format and contains simulated or measured data characterizing stand level management programmes. The forest models that are required to simulate forest development in dependence of management and climate scenarios are not part of the AFM ToolBox. The models provide the input for the database about the performance of different forest management alternatives either directly as output of forest models or via linker functions establishing a relationship between forest model output and suitable ecosystem service indicators. Such raw data are transferred to the DataBase by the AFM ToolBox client (see Fig. 10.1). The client is highly customizable and has the ability to handle the outputs of a diverse set of forest models (LandClim [24], PICUS [25], GOTILWA [7], FinnFor [20], 3PG calibrated for Portugal [9]). The data format defines a set of possible forest state and flow attributes as well as metadata

Table 10.1 Data and metadata types for the AFM ToolBox DataBase

Data type	Description
Site type	Description of site properties such as soil type, nutrient and water supply
Climate	Characterization of the used climate scenario including basic climatic averages
Stand type	Describes initial forest stand condition (species composition, silvicultural system, age, etc.)
Management	Description of the applied management concept including the regeneration phase
Forest state	Time series of indicators related to the forest state. Examples are the standing timber, biomass, carbon storage in the soil, but also indicators such as species diversity
Forest flow	Time series of indicators related to the flows from and to the forest stand (e.g. annual increment, timber harvests, tree mortality, carbon sequestration)

The forest state, flows and activities are related to actual simulation results of forest models, while the other types are related to context metadata

providing context information to the numerical simulation outputs. The data format of the AFM ToolBox DataBase contains two types of data: (1) it stores the indicators describing the development of the simulated forest stands. The time resolution is annual or lower (e.g. 10 year periods), and the data is on stand and/or species level. (2) it includes metadata providing context information to the numerical simulation data (see Table 10.1).

The DataBase viewer provides a way to inspect the content of the DataBase, the user can display simulation results graphically by plotting the various output variables over time.

10.3.2 Content Management System and Knowledge Base

The CMS is providing access to the AFM knowledge base and is guiding through the tools of the ToolBox. The knowledge base consists of content coded in html and includes a description of the forest management planning process [19, 22, 34], a collection of frequently asked questions (FAQs) about adaptive forest management and a set of case studies. Information is tagged based on two archetypical user types [manager/analyst]. These user types refer to the different information needs and user demands. The CMS interprets the tags and allows presenting the content of the knowledge base in user specific form.

10.3.3 Tools

The Optimized Management Plan (OMP) and the Vulnerability Assessment Tool (VAT) are considered as the main system elements to analyze the data of the

ToolBox Database. The OMP allows to find an optimal management plan for a given landscape. One management option is assigned to each stand entity in order to optimize the objective function at landscape level while meeting the constraints. The set of management options and their outcomes for each stand have been uploaded by the user and are stored in the DataBase. The model generator component is designed to formulate mixed integer programming (MIP) representations of forest management problems. It is prepared to read outputs from the simulations stored in the database (e.g. harvest volumes, various ecosystem service indicators) and financial data provided by the user (i.e. interest rate, prices and costs) to compute the coefficients of all variables in all equations in the model (objective function, the accounting variables and the constraints). The structure of the files that store the MIP matrices was designed to comply with the requirements of the MIP solver. The module is linked to external solvers (the GLPK open source MIP solver is used by default but the module is also ready to use Cplex).

The Vulnerability Assessment Tool allows to evaluate the vulnerability of management alternatives according to the dimensions sensitivity and adaptive capacity (compare Seidl et al. [28]). The sensitivity is expressed by indicators representing a set of ecosystem services, which are directly retrieved from the DataBase. They allow to assess the impacts of a changing climate by indicating the difference between indicator values under baseline climate and the respective value under climate change conditions. The indicators for adaptive capacity are qualitative and represent the likely institutional and financial support in a given decision making context and must be provided by the user via the GUI.

10.4 System Development

The AFM Toolbox was designed, developed and implemented in the context of the European FP6 Project MOTIVE (Models for adaptive forest management). This allowed to involve several European partners in the system development and collaborate in a sequence of steps to finalize the first prototype of the decision support system:

1. A literature review on approaches to decision making with particular focus on adaptation to climate change allowed to conceptualize generic processes of planning and decision making, vulnerability assessment schemes and uncertainty issues in climate change adaptation. Especially the needs of the target users (manager/analyst) were defined.
2. Model-based and data -driven approaches to support adaptive forest management were assessed for their potential to support the AFM ToolBox development. Here the focus was to secure the functionalities required to transfer model output into a common DataBase by means of a DataClient. As a prerequisite a common data base structure was designed to support the use of a harmonized model output.

3. The system architecture was defined to describe the relation between the main components of the ToolBox: DataBase, Vulnerability Assessment Tool, Optimized Management Plan and the content management system in collaboration with the different partners of the project.
4. Based on the definition of the user requirements and the system architecture the technical design specification was described. This supported the programming of the first prototype 0.1 of the AFM ToolBox.
5. Programming, testing and debugging of the prototype 0.1 of the AFM Toolbox was done again in collaboration with the project partners in order to revise the user interface, functionalities and handling of the different components. Through standardized questionnaires it was possible to provide a formalized feedback on the processes and release the prototype 1.0.

The core DSS development team consisted of three teams (Fig. 10.2). BOKU had the lead and cooperated closely with IFER (a SME in Prague, Czech Republic) and Satzwerkstatt, a private consultant company in Vienna, Austria. While BOKU was mainly responsible for system design in general and for defining the interfaces between the different software components, IFER had its development focus on design and implementation of DataBase and DataViewer. Satzwerkstatt, specialized in web-design and implementation, carried the main workload in implementing the DSS-frame and in designing the major GUI features. ISA from Lisbon, Portugal contributed the core functionalities of the optimizer for the OMP tool. An array of other international partners provided bits and pieces of knowledge base and case study materials. From a project management perspective it is a demanding challenge to produce a product such as the AFM ToolBox in a dispersed development environment. Being aware of potential pitfalls such as communication barriers due to different cultural and technical background of involved partners, and conflicting interests with regard to the envisaged product, the approach taken within MOTIVE tried to avoid as many problematic issues as possible. Reasons why the development process went extremely smooth were: (1) partners were used to cooperate from earlier research projects, (2) frequent physical meetings with sufficient time for

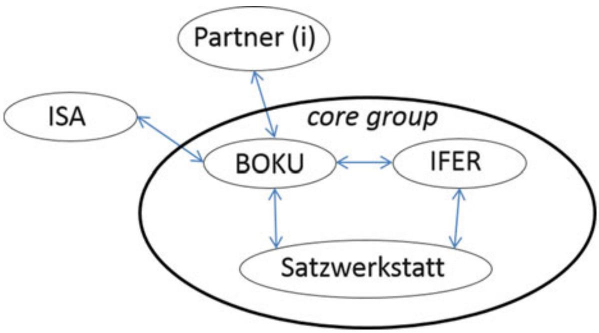


Fig. 10.2 Scheme of DSS development teams. *Arrows* indicate major flows of information

discussion among the partners, (3) clear lead role of BOKU in the development of the conceptual framework and also in the technical design process. This kept "transaction costs" of translating concepts into technical measures at a minimum.

The details on the technical implementation of the AFM toolbox are described in Sect. 10.6.

10.5 User Interface Design

The CMS of the AFM ToolBox is the central starting point providing access to the AFM knowledge base and the means to start the tools of the ToolBox (Fig. 10.3). A set of case study examples comprises several regional data sets from all over Europe to be explored by the user as a demo. The FAQs aim at the most relevant aspects of climate change (e.g. What will the future climate look like?) and refer to the contents of the DataBase.

The use of the Vulnerability Assessment Tool is split into three general steps: (1) The cases for the analysis are selected based on the available metadata in the data base. The available data can be further explored in geographical or in biophysical

Fig. 10.3 Main page of the CMS of the AFM ToolBox provides access to the knowledge base (here: "*Adaptive Management*" and "*Examples*")

space using an integrated map or via diagrams. (2) The user selects relevant ecosystem services (e.g., timber production, Carbon sequestration, biodiversity, risks) and assigns weights reflecting the relative importance of the respective indicators. (3) The analysis of the results is organized along four pre-defined questions (e.g. What is the impact of climate change, when the business-as-usual forest management is continued?). An impact bar displays the results for the chosen question for the selected cases, the respective management and climate scenario, and the selected time period on a scale from −1 (red) to +1 (green), where +1 indicates fully preferable outcomes. Changing the selection (e.g., switching between the time periods) causes animated transitions of the diagram (Fig. 10.4).

The OMP is considered an expert tool as the interpretation of the solution proposed by the optimization technique requires understanding of the methodology. However, the graphical user interface allows an easy handling and supports three basic steps in the optimization process: (1) The data set for the analysis is selected from a list of available data in the data base. Data sets comprise of a number of stands which may represent a part of the landscape and includes simulation outputs of all available stand treatment programmes for each stand. (2) The parameters for the optimization process are provided and the tool generates the corresponding MIP model and runs the solver (i.e. performs the optimization). The parameters for

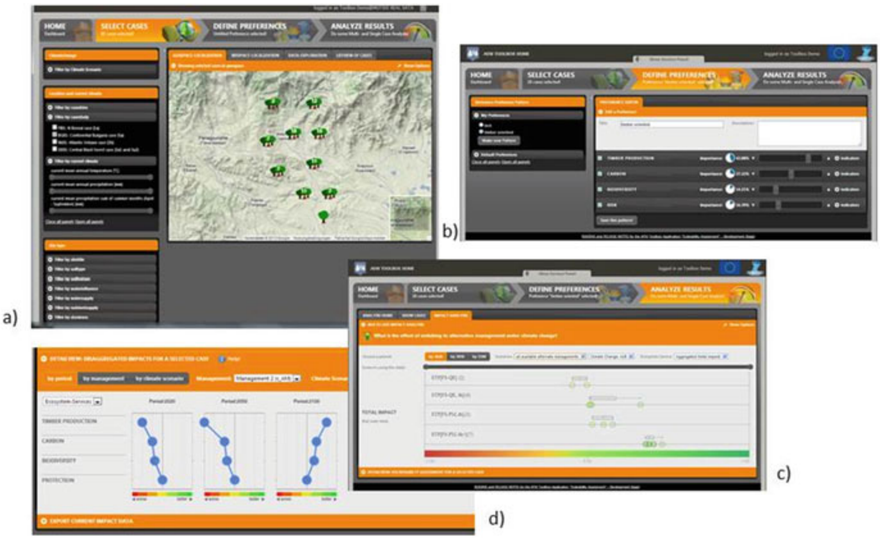


Fig. 10.4 Screens from the main steps of the vulnerability assessment tool. (**a**) Selecting the cases from a list or from maps, (**b**) selection of the ecosystem service indicators and determining their weights. (**c**) The *impact bar* shows the expected impact of climate change on ecosystem services under different management scenarios (*rows*). The *green colours of the circles* indicate a preferable outcome. (**d**) Convertible performance profiles over time periods (as shown in this screenshot), management alternatives, or climate scenarios provide a quick graphical overview on analysis results

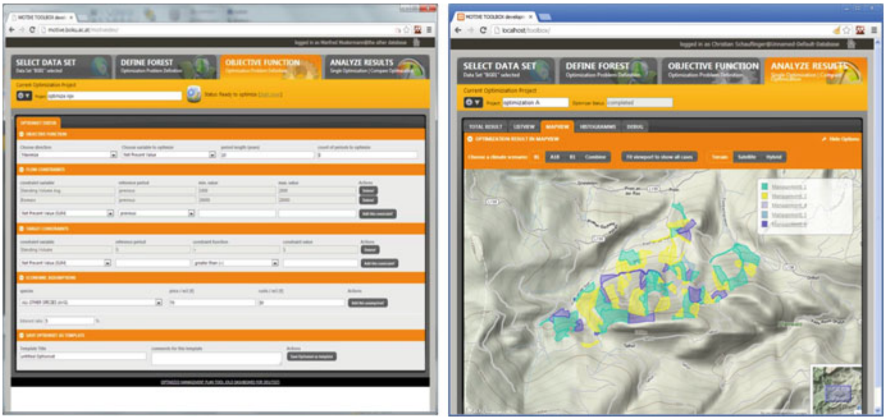


Fig. 10.5 Definition of the objective function, the flow and target constraints as well as several economic parameters for the optimization (*left*).The optimal assignment of forest management prescriptions can be viewed on a map (*right*)

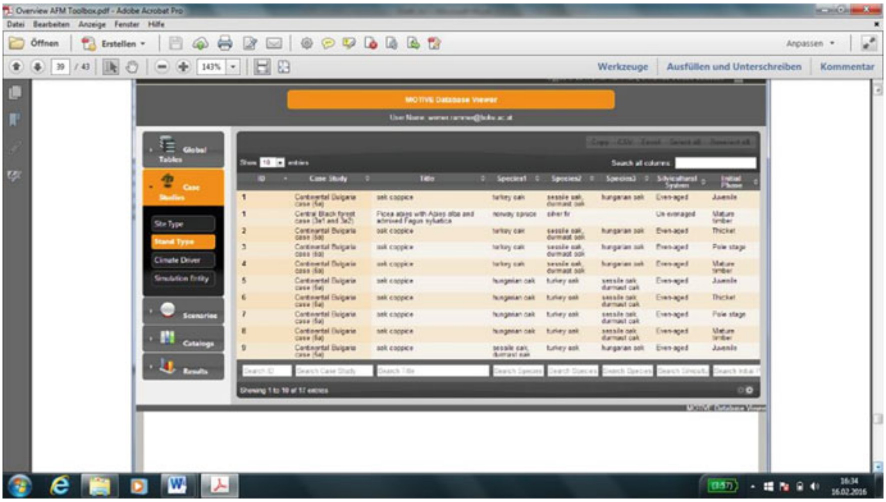


Fig. 10.6 Database viewer allows to examine the data sets, preview data and display aggregated results for selected parameters

the optimization consist of specifying the objective function as well as flow and target constraints. Additionally, the user can specify the length and number of time periods as well as economic parameters such as interest rate or harvesting costs and revenues. (3) After successfully executing the optimization, the results can be viewed and analyzed (Fig. 10.5).

Using the Database viewer the user can display simulation results graphically by plotting the various output variables over time (Fig. 10.6). Different types of charts are supported (e.g. Timber volume over time, Stems per species). All data from previous simulation runs as well as the accompanying meta data can be viewed and exported for further analysis.

10.6 System Implementation

Applying ecosystem models to generate scenario data of forest development under climate change requires high technical skills. Thus, integrating such simulation tools in a DSS which aims at practitioners as end users would be not appropriate. Analysts, on the contrary, are able to handle complex models. Therefore, a harmonized data model and a DataClient were implemented to convert the outputs of different models to the ToolBox data format. The AFM ToolBox client has built-in scripting support allowing the re-use of custom code for specific forest models. So the data driven tools of the AFM ToolBox can be utilized in different settings (Fig. 10.7). The AFM ToolBox provides demo datasets from previous projects and supports the use of customized data i.e., data that is generated by the user or for the user. In all three cases, the ToolBox is accessed via an internet browser from the users' local PC.

The technical implementation of the AFM ToolBox builds upon on a number of open source technologies which are frequently used for web development (see Table 10.2). They provide the technical foundation for the implementation of the individual tools of the ToolBox. A set of application programming interfaces (API) is used for database access and database handling, for the administration of user accounts and for persistent storage of tool-specific settings. In addition, the framework comes with a set of user interface controls allowing for a consistent visual appearance across multiple tools of the ToolBox.

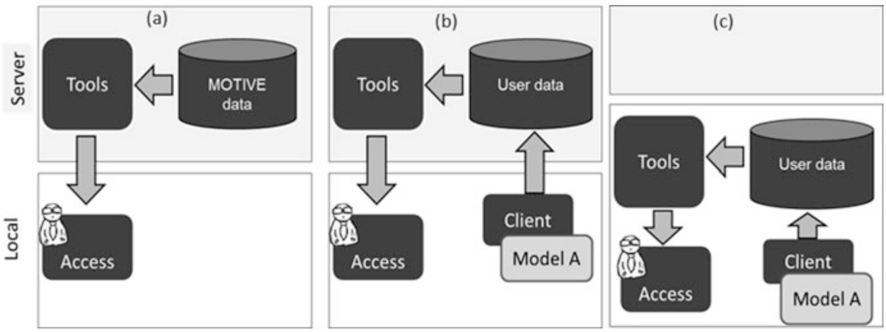


Fig. 10.7 The AFM ToolBox provides three different usage modes with regard to the location and type of the data. In case (**a**) ready demonstration data is located on the central server, (**b**) uses the central AFM ToolBox infrastructure, but with user generated data, (**c**) is a local installation with both the data and the tools running locally

Table 10.2 The AFM ToolBox is implemented based on numerous open source technologies

Technology	Description
Webserver (Apache)	Standard web server software on Linux or Windows base operating systems (http://www.apache.org/)
MySQL	Server side data base engine (http://www.mysql.com/). Used for storing simulation result data and tool specific data
PHP	Server side programming language (http://php.net/)
jQuery	Client side Javascript library (http://jquery.com/) used for the user interface
processing.js	Javascript library for visual programming (http://processingjs.org/), used for interactive diagrams
Google Maps API	Mapping technology (https://developers.google.com/maps/) used for maps display
WordPress	Content management system used for the AFM ToolBox website (http://wordpress.org/)

10.7 System User Experience

The AFM ToolBox has been tested in different environments and by various endusers during workshops focusing on DSS application and use. In two workshops feedback was provided by different users (experts, students, researchers) by standardized questionnaires. In total 32 users were asked to evaluate the AFM ToolBox with regard to criteria like the "use of the system", "look and feel", "understandability", "applicability of the tools" and "access of help and documentation".

The users evaluated positively that information is offered in various forms (background information, maps, images, documents, examples, FAQs). However, the user cannot customize or enhance the currently available contents of the knowledge base. So far in designing the functionalities of the ToolBox no authoring tools have been considered as for the online version no maintenance and quality assurance could be granted.

The look and feel of the AFM ToolBox was mostly rated as good and very good (Fig. 10.8). The handling of the different functionalities, the integration of the DSS in the CMS and the visualisation of the results were acknowledged. However, as the complexity increases a lot of information has to be communicated, which might overwhelm some users.

The evaluation has shown that most of the users (representing the "analyst" user type) have a clear understanding of the methodology used to evaluate the management options (Fig. 10.8). The selection of data sets and the exploration of different cases with the Database viewer is easy and the application of the VAT and OMP tool seems to be straight forward. Sometimes users complained about the long waiting time until results are displayed in cases where a complex calculation had to be processed on the server.

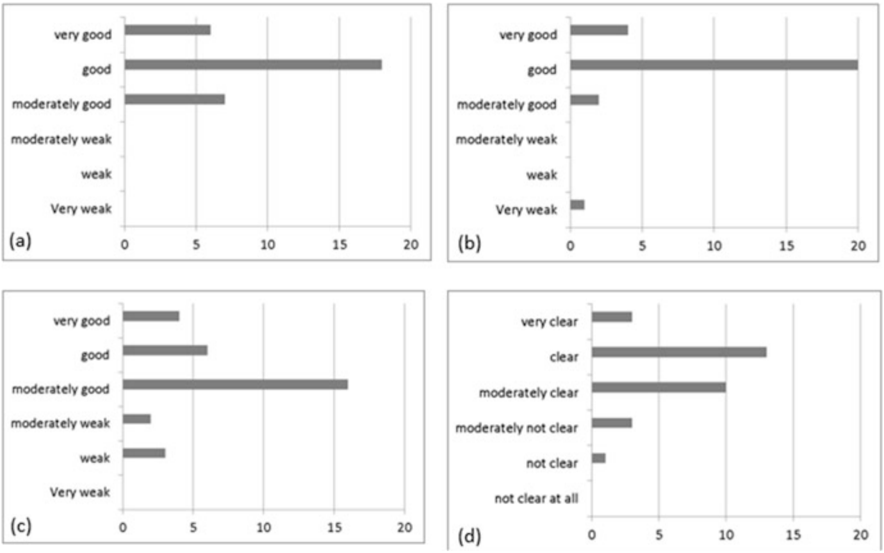


Fig. 10.8 User feedback on the look and feel of the software application (**a**), the usefulness of the documentation and help (**b**), the general understanding of the methodology (**c**) and the results displayed (**d**)

Table 10.3 lists the tools and major functionalities of the current version 1.0 of the AFM ToolBox and compares them according to a set of evaluation criteria based on Vacik et al. [33]. The Vulnerability Assessment Tool (VAT) has enormous power in the Analyst version, and provides easy to use multi-criteria assessment options in the Manager's variant. The group decision making version of the VAT is available in local installation mode only and requires an expert as facilitator. This tool has been designed to gather and combine preferences of various stakeholders and to create new knowledge by ranking management alternatives according to stakeholder preferences. The Optimized Management Plan tool (OMP) provides easy access to advanced mixed integer programming methodology with a powerful solver, however, it is actually a tool for analysts and not for practitioners. The way how facts and scenario analysis output are communicated in textual and graphical form is decisive for the acceptance by the user. The AFM ToolBox focuses on relatively simple graphical representation where the user can shift between several graphical variants to explore effects of climate and management on ecosystem service performance. This was also acknowledged by the trial users in their feedback in the questionnaires.

Free accessibility via the internet can be seen as a huge advantage in transferring state of the art information and tools to end users. However, this flexibility and ease of access comes to the cost of not having own specific data in the DataBase. To overcome this situation a user needs to run models on own data and upload them to the DataBase, either on the server or after customizing the AFM ToolBox on

Table 10.3 Self-evaluation of major AFM ToolBox functionalities (+ = criterion met, − = criterion not met, +/− = undecided, na = not applicable)

Criterion	Content management system	Database viewer	Vulnerability assessment tool	Optimized management tool
Actively create new knowledge	−	na	+	+
Making knowledge available	+/−	+/−	−	−
Increasing transparency	+/−	+	+	−
Gathering interests	na	na	+	+/−
Requires less time in applying	+	+	+	+
Low level of expertise needed	+	+	+	−
Flexibility/adapted to different needs	+/−	na	+/−	+
Helps to explore/handle uncertainty	+/−	na	+	−

the desktop computer. Here, at the latest it becomes apparent that it is not realistic to assume that all available tools can be operated by a forest manager. However, accessability of tools for forest management planning has been greatly enhanced and consultants could easily take on the role as facilitators for forest managers.

10.8 Conclusions

To promote the idea of an adaptive management approach ample emphasis is on the linkage of the tools and the knowledge base on one hand and the adaptive management cycle on the other. By linking tools and knowledge base to the operational planning and decision making processes the AFM ToolBox is promoting the quality of decision making. Through a better understanding of the pro's and cons' of different management options the transparency of the process is increased and decisions can be better justified. This general conclusion on the usefulness of DSS development has been drawn by several authors [4, 23]. However, it does not touch on the inherent dilemma of DSS development. A big advantage of the AFM ToolBox is the possibility to access the tool and the provided knowledge via the internet—the interested user can immediately try the tools and explore the decision space with available data. However, this causes also a major limitation as the user is not having own data in the database without additional efforts. A user would need to run forest models on own data and to upload them to the database, either on the server or after customizing the AFM ToolBox on the own desktop computer.

If technical complexity of a decision support process is high (i.e. the use of complex ecosystem models, multi-model simulations, spatial optimization) the cost of technical tool integration and standardization may be prohibitive for a computerized DSS either because the resources and know-how for implementation are not available, or because the use of such advanced tools is too complicated for most potential users. Thus, even if tools are available in the science labs, the transfer into practice via a DSS may be a challenge in itself.

If the procedural complexity of a decision support process is high (e.g. group mode of the Vulnerability Assessment Tool) a facilitator may be required to fully utilize the potential of the tool. These two perspectives, technical and procedural complexity, link back to the initial challenge of identifying the DSS user. We strongly believe that several user profiles need to be considered when developing advanced DSS. For the AFM ToolBox we distinguished the forest manager and the analyst as target users.

The need to focus on targeted audiences for developing successful DSS applications will force decision analysts and researchers to tailor DSSs to end-user needs. The increasing trend toward simple applications and modularity of tools will support the improved design of new DSS architectures focusing on a collection of loosely coupled tools rather than developing a single highly integrated DSS. Combining various decision support tools that support different phases of the decision making process and meet different user demands will become therefore an important feature of future DSS projects. The exchange of experiences and lessons learned from the development and application of DSS is therefore becoming more and more important. The Community of Practice of Forest Management Decision Support Systems has a well-established user community from research, public bodies, business and NGOs which allows therefore an ongoing discussion on the latest development trends [8].

References

1. Ambrose-Oji, B., Tabbush, P., Carter, C., Frost, B., Fielding, K.: Public Engagement in Forestry: A Toolbox for Public Participation in Forest and Woodland Planning. Forestry Commission, Edinburgh (2011)
2. AR5, I.W.: Climate Change 2014: Impacts, Adaptation, and Vulnerability, vols. I and II. http://ipcc-wg2.gov/AR5/report/ (2014). Accessed 20 May 2016
3. Assessment, M.E., et al.: Ecosystems and Human Well-Being: Synthesis. Island Press, Washington (2005)
4. Borges, J., Eriksson, L.: Foreword: special issue: decision support systems for sustainable forest management. Scand. J. For. Res. **29**, 1 (2014)
5. Borges, J.G., Nordstrom, E., Garcia-Gonzalo, J., Hujala, T., Trasobares, A.: Computer-based tools for supporting forest management. The experience and the expertise world-wide. Dept. of Forest Resource Management, Swedish Univ. of Agricultural Sciences (2014)
6. Bunnell, F.L., Boyland, M.: Decision-support systems: it's the question not the model. J. Nat. Conserv. **10**(4), 269–279 (2003)

7. Cotillas, M., Sabate, S., Gracia, C., Espelta, J.M.: Growth response of mixed mediterranean oak coppices to rainfall reduction: could selective thinning have any influence on it? For. Ecol. Manag. **258**(7), 1677–1683 (2009)
8. DSS, C.F.: Community of practice of forest management decision support systems. http://www.forestdss.org (2016). Accessed 20 May 2016
9. Fontes, L., Landsberg, J., Tomé, J., Tomé, M., Pacheco, C.A., Soares, P., Araujo, C.: Calibration and testing of a generalized process-based model for use in Portuguese eucalyptus plantations. Can. J. For. Res. **36**(12), 3209–3221 (2006)
10. Funtowicz, S.O., Ravetz, J.R.: The worth of a songbird: ecological economics as a post-normal science. Ecol. Econ. **10**(3), 197–207 (1994)
11. Heinimann, H.R.: A concept in adaptive ecosystem management an engineering perspective. For. Ecol. Manag. **259**(4), 848–856 (2010)
12. Kangas, J., Kangas, A.: Multiple criteria decision support in forest management the approach, methods applied, and experiences gained. For. Ecol. Manag. **207**(1), 133–143 (2005)
13. Keeton, W.S.: Managing for late-successional/old-growth characteristics in northern hardwood-conifer forests. For. Ecol. Manag. **235**(1), 129–142 (2006)
14. Kolström, M., Lindner, M., Vilén, T., Maroschek, M., Seidl, R., Lexer, M.J., Netherer, S., Kremer, A., Delzon, S., Barbati, A., et al.: Reviewing the science and implementation of climate change adaptation measures in European forestry. Forests **2**(4), 961–982 (2011)
15. Lexer, M., Vacik, H., Palmetzhofer, D., Oitzinger, G.: A decision support tool to improve forestry extension services for small private landowners in southern Austria. Comput. Electron. Agric. **49**(1), 81–102 (2005)
16. Lexer, M.J., Seidl, R.: Addressing biodiversity in a stakeholder-driven climate change vulnerability assessment of forest management. For. Ecol. Manag. **258**, S158–S167 (2009)
17. Lindner, M., Maroschek, M., Netherer, S., Kremer, A., Barbati, A., Garcia-Gonzalo, J., Seidl, R., Delzon, S., Corona, P., Kolström, M., et al.: Climate change impacts, adaptive capacity, and vulnerability of European forest ecosystems. For. Ecol. Manag. **259**(4), 698–709 (2010)
18. Liu, S., Innes, J., Wei, X.: Shaping forest management to climate change. For. Ecol. Manag. **300**, 1–3 (2013)
19. Mintzberg, H., Raisinghani, D., Theoret, A.: The structure of “unstructured” decision processes. Adm. Sci. Q. **21**, 246–275 (1976)
20. Pyörälä, P., Kellomäki, S., Peltola, H.: Effects of management on biomass production in Norway spruce stands and carbon balance of bioenergy use. For. Ecol. Manag. **275**, 87–97 (2012)
21. Rammer, W., Schauflinger, C., Vacik, H., Palma, J.H., Garcia-Gonzalo, J., Borges, J.G., Lexer, M.J.: A web-based toolbox approach to support adaptive forest management under climate change. Scand. J. For. Res. **29**(suppl. 1), 96–107 (2014)
22. Rauscher, H.M.: Ecosystem management decision support for federal forests in the united states: a review. For. Ecol. Manag. **114**(2), 173–197 (1999)
23. Reynolds, K.M., Borges, J.M., Vacik, H., Lexer, M.J.: Information and communication technology in forest management and conservation. In: Hetemäki, L., Nilsson, S. (eds.) Information Technology and the Forest Sector. IUFRO World Series, vol. 18, pp. 150–171. International Union of Forest Research Organizations, Vienna (2005)
24. Schumacher, S., Bugmann, H., Mladenoff, D.J.: Improving the formulation of tree growth and succession in a spatially explicit landscape model. Ecol. Model. **180**(1), 175–194 (2004)
25. Seidl, R., Lexer, M.J., Jäger, D., Hönninger, K.: Evaluating the accuracy and generality of a hybrid patch model. Tree Physiol. **25**(7), 939–951 (2005)
26. Seidl, R., Rammer, W., Jäger, D., Currie, W.S., Lexer, M.J.: Assessing trade-offs between carbon sequestration and timber production within a framework of multi-purpose forestry in Austria. For. Ecol. Manag. **248**(1), 64–79 (2007)
27. Seidl, R., Rammer, W., Jäger, D., Lexer, M.J.: Impact of bark beetle (Ips typographus L.) disturbance on timber production and carbon sequestration in different management strategies under climate change. For. Ecol. Manag. **256**(3), 209–220 (2008)

28. Seidl, R., Rammer, W., Lexer, M.J.: Climate change vulnerability of sustainable forest management in the eastern alps. Clim. Chang. **106**(2), 225–254 (2011)
29. Twery, M.J., Knopp, P.D., Thomasma, S.A., Rauscher, H.M., Nute, D.E., Potter, W.D., Maier, F., Wang, J., Dass, M., Uchiyama, H., et al.: Ned-2: a decision support system for integrated forest ecosystem management. Comput. Electron. Agric. **49**(1), 24–43 (2005)
30. Vacik, H., Lexer, M.J.: Past, current and future drivers for the development of decision support systems in forest management. Scand. J. For. Res. **29**(supp. 1), 2–19 (2014)
31. Vacik, H., Lexer, M.J., Palmetzhofer, D., Stampfer, K., Limbeck-Lilienau, B.: Anwendung der entscheidungshilfe cones zur planung von verjüngungseingriffen im gebirgswald. In: Strobl, J., et al. (eds.) Angewandte Geoinformatik 2004, Beitrge zum 16. AGIT-Symposium Salzburg, Wichmann, Heidelberg (2004)
32. Vacik, H., Lexer, M.J., Rammer, W., Seidl, R., Hochbichler, E., Strauss, M., Schachinger, C.: ClimChAlp-a webbased decision support system to explore adaptation options for silviculture in secondary norway spruce forests in Austria. In: Falcao, A., Rosset, C. (eds.) Proceedings of the Workshop on Decision Support Systems in Sustainable Forest Management, Lisbon. OP3, Lisbon, Portugal (2010)
33. Vacik, H., Borges, J.G., Garcia-Gonzalo, J., Eriksson, L.O.: Decision support for the provision of ecosystem services under climate change: an editorial. Forests **6**(9), 3212–3217 (2015)
34. Walters, C.J., Holling, C.S.: Large-scale management experiments and learning by doing. Ecology **71**(6), 2060–2068 (1990)
35. Wang, S.: One hundred faces of sustainable forest management. For. Policy Econ. **6**(3), 205–213 (2004)

Chapter 11
SINGRAR—A Distributed Expert System for Emergency Management: Context and Design

Mário Simões-Marques

Abstract Complex Emergency Management operations require strong capabilities and robust decision-making support by means of Intelligent Systems. The chapter describes the problem context, the underlying theoretical concepts, system requirements and architecture, knowledge management process as well as the spiral development approach adopted in the design of the SINGRAR expert system to dynamically manage response priorities on emergency situations, based on situational parameters. SINGRAR is a knowledge-driven DSS that was implemented in a customizable distributed shell that was developed to be scalable and adaptable to different scenarios. The first implementation of the system became operational over 10 years ago and addressed the management of critical incidents onboard of Navy ships. Meanwhile the system was customized to different classes of ships and received continuous improvements, both in terms of functionality and usability, which are addressed in different dimensions.

11.1 Introduction

11.1.1 General Considerations

Emergency Management is a complex process that requires coordination of different actors, with different cultures, aims and views of the world. The development of decision support systems (DSS) that support the decision-makers and provide a common picture for crises response operations is a big challenge.

SINGRAR is the Portuguese acronym for Priority Management and Resource Assignment Integrated System. According to the classification presented earlier in this book SINGRAR is a knowledge-driven DSS. This system was developed

M. Simões-Marques (✉)
Centro de Investigação Naval (CINAV) – Portuguese Navy, Alfeite,
2810-995 Almada, Portugal
e-mail: mj.simoes.marques@gmail.com

© Springer International Publishing Switzerland 2016
J. Papathanasiou et al. (eds.), *Real-World Decision Support Systems*,
Integrated Series in Information Systems 37,
DOI 10.1007/978-3-319-43916-7_11

to dynamically manage response priorities on critical situations based on situational parameters, and to advice on resource assignment considering, for instance, individual capabilities and location. SINGRAR core Fuzzy Multiple Attribute Decision Making model was implemented in a customizable distributed shell that was developed to be scalable and adaptable to different scenarios. The system became operational a decade ago and addressed the management of critical incidents onboard of Navy ships, tackling equipment repair and damage control advice considering multithreat battle and emergency environments. Meanwhile the system was customized to different classes of ships and received continuous improvements, both in terms of features and usability. Particularly regarding SINGRAR usability, an extensive study was conducted assessing its quality and identifying areas of improvement.

Ship emergencies tend to have catastrophic impacts namely in terms of human casualties, economical losses and/or environmental damage, particularly when they are unsolved or ineffectively solved. It is still fresh in our memories the accident with the MS Costa Concordia cruise ship which, in January 2012, hit a rock off the island of Giglio, Italy with a death toll of 32. Maritime accidents are more common than most people would anticipate. For instance, a recent European Maritime Safety Agency's review on maritime accidents in and around European Union waters reported a total of 9180 occurrences (ranging from marine incidents at the lower end of the scale through to very serious accidents) in the period of 2011–2014, causing the loss of more than 390 lives and about 3250 injured [8]. In 2014 alone, the European database on marine accidents records 3025 accidents, 99 of which were very serious accidents, involving 3399 ships, resulting in the loss of 51 ships, in 136 fatalities, and 1075 persons injured. The same review refers that the data stored in the European Marine Casualty Information Platform (EMCIP) by the accident investigation bodies of the EU Member States suggests that approximately 3500 occurrences could be expected to be notified annually. The severity of the situation around the globe is similar.

Besides merchant shipping, Navy ships are also exposed to emergencies. The most logical situations that may cause emergencies in warships are the ones resulting from warfare activity. Besides battle incidents resulting from opponent forces' interactions during wars, which are quite infrequent nowadays, there are plenty of other examples of emergencies, namely some resulting from unprovoked or terrorist attacks. One can refer three examples of such incidents that were widely covered by media involving the USS Stark (on 1987), the USS Cole (on 2000), and the ROKS Cheonan (on 2010). USS Stark was struck by two Exocet missiles fired by an Iraqi aircraft on May 1987, during the Iran-Iraq War; in this incident 37 sailors were killed and 21 were injured, and the ship was heavily damaged. USS Cole was the target of an Al-Qaeda attack on 12 October 2000, in the port of Aden, Yemen; 17 sailors were killed, 39 were injured, and the ship was heavily damaged. ROKS Cheonan was sunk on 26 March 2010 off the Republic of Korea west coast, when carrying 104 personnel; in this incident 37 sailors were killed; an international investigation concluded that the warship was sunk by a North Korean torpedo fired by a midget submarine.

The human element is commonly recognized as a critical asset in ensuring the safety of all actors in the maritime environment both regarding the prevention of critical situations and the remediation of the ones that occur. For instance, in November 1997 the International Maritime Organization Assembly adopted a resolution on the human element vision, principles and goals for the Organization, which was updated at the end of 2003 [12]. This document points to the need of increasing the promotion of a maritime safety culture, crew performance (which is affected by individual capabilities, management policies, cultural factors, experience, training, job skills, work environment, among other factors), and dissemination of information, a key element for sound management and operational decisions.

Simões-Marques and Nunes note that despite experienced professionals can dispense decision support for routine tasks, decision support tools are a major asset and provide competitive advantage particularly for real live management of complex and high stress situations, where humans tend to fail their judgments [25]. In the maritime context the decision-making complexity depends on the size, type and activity of the ship and is affected by the risks and conditions inherent to operating at sea, which is often a very adverse environment. To better understand the rational for selecting the problem discussed in this paper one has to reckon that military ships tend to present more risks and complexity than merchant ships. On one hand, this is due, for instance, to the variety and type of equipment operated (e.g., high power electromagnetic transmitters, weapons, aircrafts) and to the type of activities performed day and night (e.g., combat operations and training, search and rescue, replenishment at sea), virtually in any weather conditions. On the other hand, by nature, military ships are designed to sustain battle damage and crews are trained to control damages and to repair ship equipment to the limit of their capabilities.

In recent years there was a growing attention devoted by the scientific community to the topic of DSS applicable to shipboard operations, particularly addressing emergency management (which includes what is named as 'damage control' in seamen terminology). Same examples of work published in this area are: Simões-Marques et al. presented a model for dealing with the prioritization of lines of action (related with ship equipment repair) in critical situations [24]; Simões-Marques and Pires further expanded this concept, discussing the SINGRAR fuzzy expert system underlying model, which added resource assignment functionalities to the prioritization of lines of action for engineering and damage control activities [23]; Lee et al. [17], Lee [16] and Calabrese et al. [4] presented proposals in the areas of damage control; and Cebi et al. [5], Vanem and Ellis [28] and Abou [1] presented proposals in the areas monitoring systems and troubleshooting. Other research efforts were focused in decision support tools designed to prevent emergency situations or to improve the response capability. Examples of references in these lines are: Kowalski et al. [14, 15], Lozowicka [19] and Arendt [2] which proposed different approaches to consider in ship system design; or Liu et al. [18] and Chauvin and Lardjane [7] that proposed methods for collision and grounding avoidance.

Despite this increase in the number of areas covered and proposed solutions, one can still identify a lack of applied research addressing holistically the problem of decision support in the context of maritime/naval emergency management, addressing both from the standpoints of response prioritization and asset assignment. The only reference found to this topic is Simões-Marques and Pires [23]. The picture does not differ substantially for the more general context of emergency management, as discussed by Simões-Marques and Nunes [25]. Considering that modern ships increasingly offer information systems to monitor and control platform status, including support to damage control, one can wonder why these systems don't offer decision-making support for these critical features. Several reasons can be pointed for this fact, related with complexity of the problem at hand, with the uncertainty and vagueness of data processed, or with the lack of "standard" methodologies to deal with the problem.

This chapter presents part of the SINGRAR case study,[1] describing the context of emergency management and offering a discussion on the main features of the SINGRAR. The chapter also addresses the system live cycle, from the concept phase until the current service phase, referring the needs, requirements, design solutions, as well as the knowledge engineering process used for the implemented solutions.

11.1.2 Problem Characterization

For the Portuguese Navy the commissioning of the "Vasco da Gama" frigates, in the early 90s, was the turning point in terms of Command and Control (C2) technologies. These ships were equipped with tactical (focused on the external environment) and platform (focused on the internal environment: ship, propulsion and power plant) command and control systems. However, these two main systems were independent and isolated, and the coordination of the needs/constrains of the two user groups was performed based on human processes and voice communications, supported by *aide mémoire* paper boards.

The interaction among the two groups is critical since the internally-focused activities—e.g., engineering, damage control, medical and logistics—support the operability and survivability of the ship, crew and systems, namely the ones that support the fighting capabilities. The liaison between these two complementary domains is performed by an organizational structure responsible for the conduction of crises response/emergency management.

Effective emergency management is a complex process that requires the coordination of different actors, with different cultures, aims and views of the world. This is particularly true for Navy ships, which are small worlds where many people, technical domains and systems intertwine performing multiple complementary and often conflicting tasks. Traditionally, coordination was assured based on a hierarchical stovepiped structure, supported by vertical communications, from decision-makers down to operators and back. This type of organization is problematic considering

[1]The remaining part of the SINGRAR case study is presented in Chap. 12.

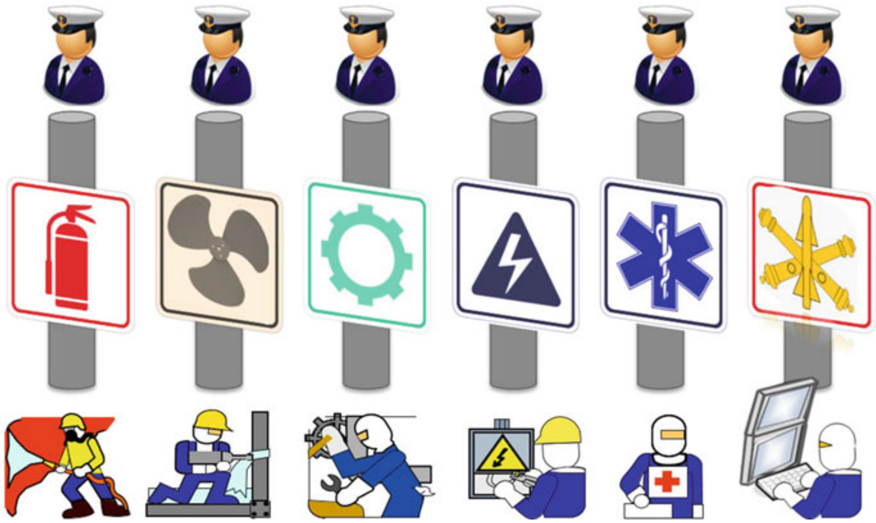


Fig. 11.1 Traditional stovepiped decision-making processes on ships

situational awareness, information sharing and unity of effort (i.e., articulation of each other's objectives), which are critical characteristics of a comprehensive, integrated and coordinated approach to emergency management. Figure 11.1 illustrates these independent processes as different stovepipes focused on specific technical or operational areas. Usually, activities' deconfliction and problem mitigation have to be performed by top level decision-makers, which are the ones that tend to have better awareness of the Command goals and a broader picture about the internal and external contexts.

The increase in number, complexity and interdependency of systems, and the need for quick response forced the evolution of the organization, methods and procedures. A pragmatic approach to tackle the problem was the creation of paper based structured information boards which tentatively combine standardized recording fields with a basic level of predefined decision-aid features. Figure 11.2 depicts part of an A0-size board used by the Weapon Engineering department of the "Vasco da Gama" frigates to record incidents with systems of its responsibility.

Depending on ships purpose and characteristics, systems architecture and warfare capabilities differ substantially. Nevertheless, independently of ship specific characteristics, in naval operations when the resources are insufficient to respond the amount of incidents some type of response prioritization is required in order to maximize operational capabilities and survivability. The above mentioned board can provide some basic decision aid, as illustrated in Fig. 11.3, which presents a table (in the lower part of the figure) that lists equipment and the corresponding repair priorities (1 to 3) based on the assumption of a 'Command Priority' (A to F) which is inferred from the combination of 'Threat Warnings' (Red to White) that are effective in the prevailing operational scenario.

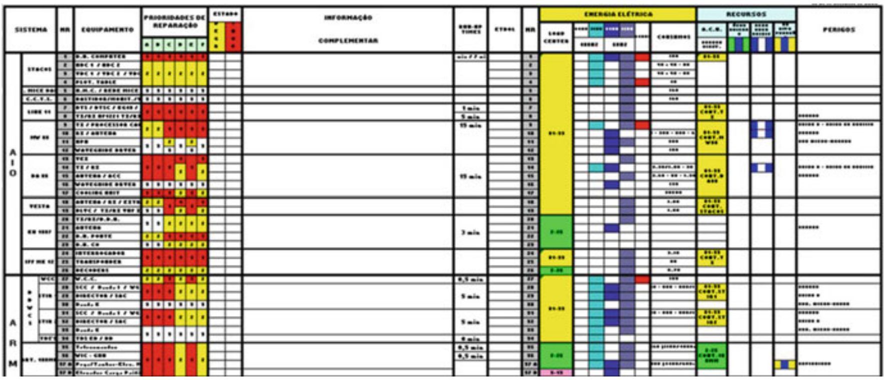


Fig. 11.2 Typical information board used to assist decision-making regarding equipment repair on Navy ships

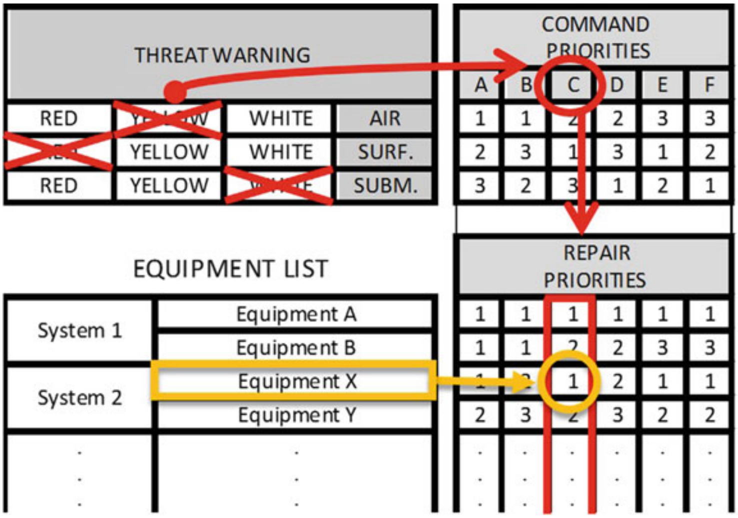


Fig. 11.3 Basic decision aid for assisting response prioritization

The reasoning associated with this method is that the importance of repairing a particular item of faulty equipment depends on its utility for the current operational environment (for instance, an anti-air missile is useful for Air Warfare but is useless for Submarine Warfare).

The main advantages of this pen and paper method are: it very simple to use; provides helpful hints to inexperienced decision-makers, particularly when no better information is available; and its use doesn't require any power or technology. However, this method has severe limitations: due to the size of the table the process is prone to errors; the process is manual and static, therefore any changes on the operational environment affecting the Command Priority force a complete and time-consuming revision of the repair priorities; the information sharing is based on voice communications, which are not totally reliable both in terms of quality and completeness of the message, therefore decision-makers located in different parts of the ship tend to have partial (and eventually incorrect) knowledge about the situation. Furthermore, for practical reasons of keeping the table manageable, it is not feasible to list all the items which are object of concern, nor other relevant information that affects the analysis and decision process. Finally, one of the most severe problems is that any decision-maker which is unaware of the limitations of the method and uses its advice blindly can be led to take completely inadequate decisions. Figure 11.4 illustrates a situation where this problem becomes evident. In the illustrated case there are two Threat Warnings at the same level, which turns the Command Priority selection process unclear. Notice that depending on the Command Priority choice (A or C) the repair priority advice for Equipment B varies (1 or 2), which may have a significant impact in recovery of the warfare capabilities that this equipment contributes to.

The Emergency Management decision-makers soon realized that an improved decision support process was required to deal with the amount of data processed, to increase the situational awareness and to strengthen the information sharing.

In fact, despite both the Warfare component and the Emergency Management component of a ship share the same type of decision cycle (shown in Fig. 11.5), which is commonly referred as the OODA loop (Observe-Orient-Decide-Act), until recently the investment in technological tools to assist these two activity domains was quite different. In effect, the need for investment on systems that integrate sensors, command and control, weapons and communication systems to support the warfare component was recognized a long time ago; while the need for investing in the same type of means for assisting the technical-logistical and emergency

Fig. 11.4 Example of problem affecting the manual/static decision aid

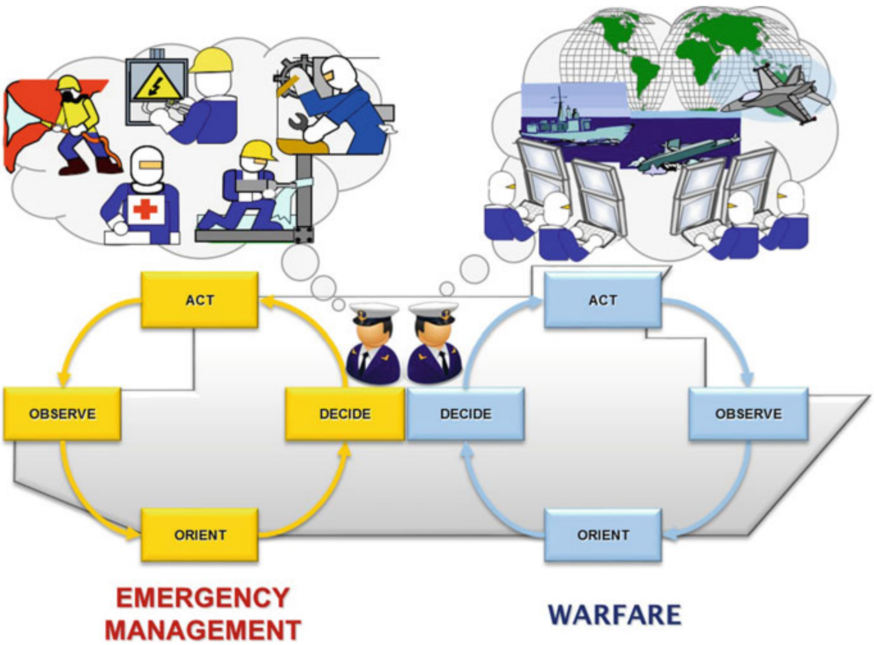


Fig. 11.5 Warfare and Emergency Management components have identical decision cycles

management activities which are critical for the operability and performance of the war fighting systems is still quite far from being well understood.

Nowadays the fulfillment of the emergency management decision-support requirements is quite feasible considering the technological evolution of computers and networks, and the emergence of new scientific methodologies (e.g., Artificial Intelligence) that paved the way for the implementation of solutions able to deal with complex problems. In our case the complexity of the problem is increased manifold, since the decision draws on a process that combines addressing the priority of a high number elements, considering a dense network of relationships and the dynamic influence of variations in the context, all this considering the variety of highly specific technical domains involved, the geographical dispersion of the actors, and the vagueness and imprecision of the data processed.

All considered it is evident that adopting a conventional approach for providing decision support (e.g., based on classical logic inference processes) is unfeasible. This problem calls for an innovative approach that, depending on the context, advices line of action priorities and the assignment of resources. Since there are many factors involved in the inference process the advice is not always obvious, therefore it is also mandatory to complement the decision advice with explanation functionalities, which help the decision-maker to validate the proposed lines of action. Furthermore, bearing in mind that the implementation of the lines of action usually requires accessing technical or logistical information, the aid process calls for further support by making available relevant contents of a knowledge base, related with the event under consideration.

As mentioned before, the decision-makers are located in different parts of the ship. Thus, the decision-support calls for a common distributed platform that offers means for sharing information, improving situational awareness, contributing to coherent and timely efforts.

SINGRAR was developed to fill this gap. The next sections address the main characteristics of the system, discuss the requirements and describe its development process.

11.1.3 SINGRAR General Characteristics

SINGRAR is a Distributed Expert System. Due both to the complexity of the evaluation and advice problems handled and to the vagueness of most of the data under consideration, a fuzzy logic [33] approach was selected for designing the inference process. Figure 11.6 depicts a high level view of SINGRAR underlying concept. The system:

1. provides a common platform for the compilation of incident status, generation of a common consolidated picture, and dissemination of advice, supporting the Emergency Management component and interfacing with the Warfare component;
2. includes decision support features that offer advice on the priorities of alternative courses of action (e.g., equipment repair, damage control) based on the current operational context (e.g., threat assessment);
3. supports the damage control C2 process;
4. advises on procedures for implementing courses of action regarding, for instance, ship's combat system, platform and damage control;
5. federates different databases (e.g., personnel, logistics, documentation) which are used in conjunction with SINGRAR specific knowledge base;
6. ensures alternative means of communication (e.g., data transfer, chat, tele- and videoconferencing) between decision-makers and responders; and
7. provides means for virtual training and for simulation.

SINGRAR was implemented in a flexible and scalable expert system shell (specifically developed for the system), which allows the parameterization of their components (knowledge base, inference engine and interface) according to the characteristics of the Universe of Discourse. For instance, when used in naval applications the ship's characteristics can be configured, thus accommodating virtually any type of ship. Nevertheless the same system can also be used for other types of applications (e.g., industrial, urban, regional) allowing the configuration of a variety of different facilities and processes, accommodating virtually any type of infrastructure and organization.

The distributed architecture has several advantages over the classical manual procedures. Some of the more relevant advantages are the instantaneous integration

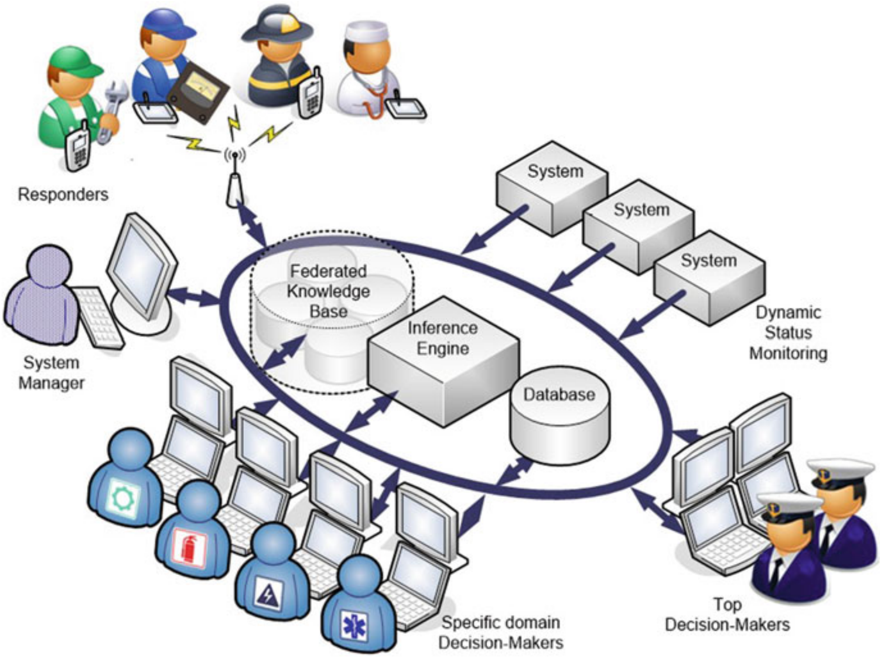


Fig. 11.6 High level view of the SINGRAR concept

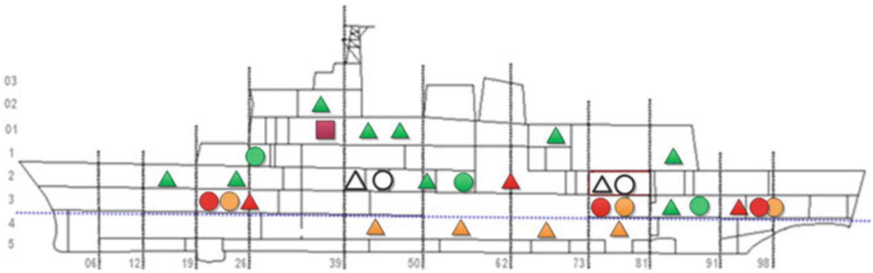


Fig. 11.7 Distribution of SINGRAR users (*square*—main decision center; *circle*—technical decision centers; *triangle*—responders' groups)

of the information compiled at different workstations; the automatic and coherent reaction to data changes; the fault tolerance; the increased survivability of compiled information and decision support capabilities; and the decrease of total time between acknowledging an incident and the triggering of the response, thus improving tempo, information sharing, situational awareness, responsiveness and coordination.

Figure 11.7 illustrates the geographical distribution of the groups of SINGRAR users. The square marks the location of the top decision-makers; the circles mark the location of decision centers of specific technical areas (color coded); while the triangles mark the location of groups of responders.

11.2 System Requirements Analysis

SINGRAR design process generically adhered to ISO recommendations for human-centered design of computer-based interactive systems. ISO 9241-210:2010 identifies four key activities (presented in Fig. 11.8), which must be performed iteratively until the solution meets the requirements [13]:

- understand and specify context of use;
- specify the user and organizational requirements;
- produce design solutions; and
- evaluate design against requirements.

This section will discuss the topics related with the first two activities of the user-centered design cycle. Sections 11.3 and 11.4 (together with Sects. 12.3 and 12.4 of Chap. 12) address the issues related with the development and implementation of the system (corresponding to the third key activity). The last activity of the user-centered design cycle (usability evaluation) is the focus of Sect. 12.4 in Chap. 12.

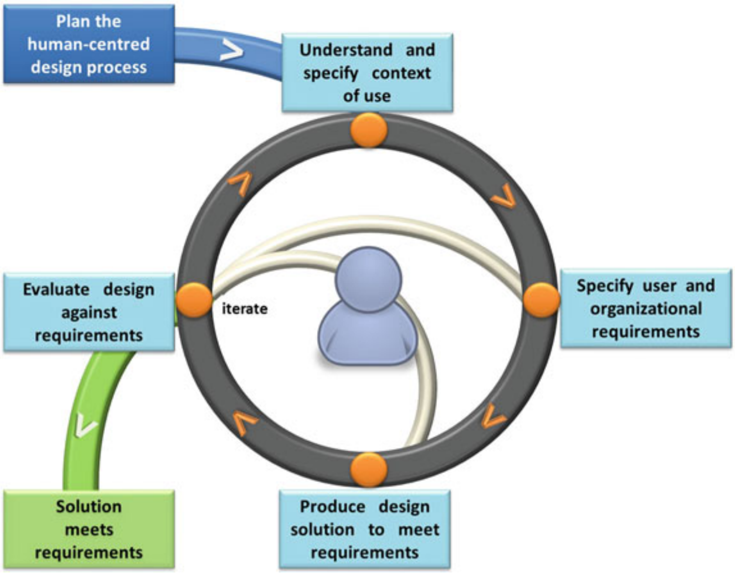


Fig. 11.8 Activities of user-centered design, adapted from ISO 9241-210 [13]

11.2.1 Context of Use

Let's first specify the context of use, characterize the users, the tasks performed and the organizational environment:

1. **Context of use**—the specific context of use of SINGRAR was broadly characterized in Sect. 11.1. It is further assumed that the system should comply and whenever possible support the general principles of Emergency Management, as defined by FEMA [10], which stated that this activity must be:

   Comprehensive emergency managers consider and take into account all hazards, all phases, all stakeholders and all impacts relevant to disasters;
   Progressive emergency managers anticipate future disasters and take preventive and preparatory measures to build disaster-resistant and disaster-resilient communities;
   Risk-driven emergency managers use sound risk management principles (hazard identification, risk analysis, and impact analysis) in assigning priorities and resources;
   Integrated emergency managers ensure unity of effort among all levels of governance and all elements of a community;
   Collaborative emergency managers create and sustain broad and sincere relationships among individuals and organizations to encourage trust, advocate a team atmosphere, build consensus, and facilitate communication;
   Coordinated emergency managers synchronize the activities of all relevant stakeholders to achieve a common purpose;
   Flexible emergency managers use creative and innovative approaches in solving disaster challenges;
   Professional emergency managers value a science and knowledge-based approach; based on education, training, experience, ethical practice, public stewardship and continuous improvement.

   Considering this, SINGRAR aims to provide efficient and effective responses to multiple and often conflicting needs in situations of scarce resources, considering several complementary functional elements, such as Supply, Maintenance, Personnel, and Health. The decision support goal is to help decision-makers answer the very basic questions What, Where, When, Who, Why, How, How Much. Despite difficult to obtain, these answers are fundamental in critical situations, where the urgency and impact of the decisions is especially sensitive, and resources are usually very limited. The use of common robust advice tools in a distributed/collaborative environment ensures predictability and coherence of the parallel and concurrent decision-making processes, which facilitates unity of effort;
2. **Users**—SINGRAR is used by Navy military personnel onboard ships, and across the entire spectrum of naval operations. Users can be grouped in four categories with different levels of interaction with the system (refer to Fig. 11.6):

Top level decision-makers interact with SINGRAR to get situational awareness, to take cross-domain decisions (e.g., take warfare decisions based on the status of ship systems, define Command Goals and Priorities based on the external environment), and to communicate with other users;

Specific domain decision-makers interact with SINGRAR to manage specific domain operations, to get situational awareness, to get and share cross-domain information, to update the common picture regarding the ship and systems status, to get advice on action and resource assignment priorities, to get explanations regarding the advice, to get further support (for instance regarding procedures and technical and logistical information), and to communicate with other users;

Responders interact with SINGRAR to receive tasking assignments, to get situational awareness, to get and share cross-domain information, to update the common picture regarding the ship and systems status, to get operational support (e.g., procedures, technical documentation, logistical information, hazards and constraints), and to communicate with other users; and

System manager interact with SINGRAR to perform configuration and troubleshooting activities (e.g., hardware, knowledge-base) and to perform user profile management;

3. **Tasks**—the tasks performed by a user depend on his/her profile, which is selected according with the responsibilities within the ship's organization. There are tasks that are common to all users (for instance, the visualization of Command Goals, threat warning and consolidated picture about the ship status) while others are only performed by particular users, according to technical domain and functional role.
4. **Operational environment**—SINGRAR can be used in all degrees of readiness of the ship, from the ship moored alongside to Battle Stations. The number of workstations required varies, increasing as the ship readiness evolves towards Emergency or Battle Stations, which is the highest. SINGRAR is installed mainly in desktop computers, but runs also in portable computers and other mobile devices. Portable computers are used, for example, in Damage Control command posts. In these posts operators are standing and working in a confined area that usually does not allow the use of an external pointing device (e.g., mouse or trackball), therefore these workstations are not suitable for an extended operation of the system. Desktop are the main type of SINGRAR workstations since they allow a more comfortable interaction with the computer (a factor which is very important to ensure an efficient, effective and satisfactory use of the application) and also because they can more easily accommodate hardware expansion requirements (e.g., memory, multiple screen graphics cards). Users frequently have to wear personal protective equipment (e.g. gloves, anti-flash gear, breathing masks) which may affect their ability to operate the system. The use of mobile devices and wireless-communications is limited considering both signal propagation and electromagnetic signature constraints.

11.2.2 User and Organizational Requirements

From a functional standpoint the main requirements for SINGRAR are:

- improved response tempo and coordination when compared with C2 procedures based on voice communications and information boards;
- timely decision support for emergency management, in a distributed environment;
- dynamically adjust advice according to context evolution;
- automatic and consistent reaction to data changes;
- instantaneous integration and consolidation of the standardized information compiled at different workstations;
- broadcast of advice on recommended lines of action and resource assignment, and prediction of operational impact resulting from incidents;
- increased survivability of compiled information and of decision support capabilities in case of system failure;
- information access based on roles and user profiles;
- reduced human workload on command and control activities;
- reduced human error;
- reduced time lag between acknowledging an incident and triggering the response;
- customizable, scalable and flexible, allowing the adaptation to different realities and user needs;
- support to virtual training and emergency simulations;
- mediate and transparently provide access to external sources of information (e.g., personnel, logistic and documental databases).

The requirements gathering and definition, and SINGRAR's current architecture were not set in a single process. In fact, SINGRAR resulted from an incremental development approach that lasted several years, which was achieved based on a combination of professional experience, scientific research and the Portuguese Navy support through the availability of funds and software developers.

This project served as test bench for the development of innovative concepts and the experimentation of new technologies. For instance, in early 00's and before RFID technology became generalized, SINGRAR architecture integrated ID card readers that were used to update the crew status when the ship was alongside (i.e., onboard or ashore), and that were also used to test the automation of the 'quick count/identification' procedure which is performed in situations of accident or 'man overboard' when the ship is sailing.

11.3 System Design

There are strong reasons to justify the lack of Intelligent Systems to support Emergency Management activities. Besides the complexity of the decision process due to the high number of parameters and relations under consideration, there is also the problem of developing reasoning mechanisms able to deal with the meaning of vague concepts usually used. For instance, for characterizing an emergency situation using natural language it is common to use linguistic expressions such as “severe damage”, “very degraded”, “quickly repaired” or “very important asset”. Even if the language is constrained by some formalism, remains the question on how to handle statements such as “equipment A, which is fundamental to respond to threat X, is degraded” or “asset B, which is very important to respond to incident Y, is unavailable”. Classical Set Theory and Boolean logics present serious limitations to manipulate data that has such ill-defined outlines.

Expert systems are a particular type of Intelligent Systems which, as Turban et al. [26] note, through the use of applied artificial intelligence techniques, aim to reach a level of performance comparable to human experts, mimicking them in a particular area. The artificial intelligence methodologies used by Intelligent Systems are diverse, including fuzzy reasoning, rule-based, case-based, evolutionary algorithms, machine learning approaches, to name a few. When compared to natural intelligence based decision-support in a specific domain, an artificial intelligence system offers some advantages since [26]: (1) it is more permanent, (2) is easy to duplicate and disseminate, (3) can be less expensive, (4) is consistent and thorough, (5) can be documented, (6) can execute certain tasks much faster than a human, and (7) can perform certain tasks better than many people.

An intelligent system is composed by four core building blocks illustrated in Fig. 11.9 that perform the following functions:

- **Knowledge Base**—stores the knowledge required for solving a specific domain problem;
- **Working Memory**—stores data or facts about the particular problem context under analysis;

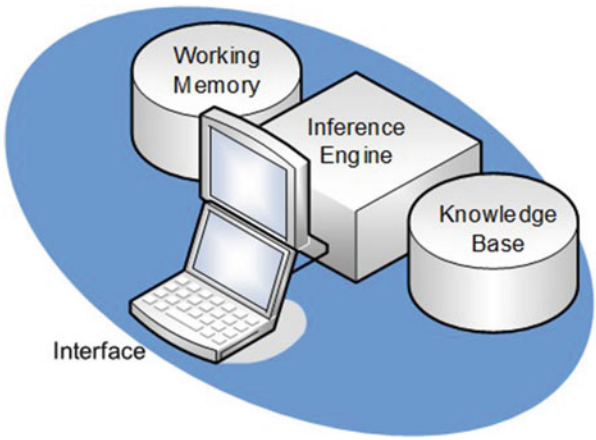


Fig. 11.9 Basic architecture of an Intelligent System

- **Inference Engine**—runs the knowledge against the data, assessing the concrete situation and generating conclusions and advise, as well as explanations;
- **User Interface**—offers the human-computer interaction means necessary to input data, insert requests and obtain system outputs.

11.3.1 Knowledge Management

As a distributed Intelligent System the main goals of SINGRAR are: providing dynamic advice to decision-makers regarding courses of action (including explanation); ensuring a reliable and flexible network to support information sharing; and serving the needs of the various actors engaged on the decision process, including the means for compiling information to produce a standardized, integrated and consolidated picture on the status of incidents and on the usage of resources.

Standardizing, integrating and consolidating information requires eliciting knowledge in problem domain (e.g., ontologies to characterize the types of incidents and the resources useful to the potential areas of emergency response) [11]. The European Committee for Standardization's "European Guide to good Practice in Knowledge Management – Part 1: Knowledge Management Framework", issued in March 2004, offers a working definition of knowledge [6]:

> Knowledge is the combination of data and information, to which is added expert opinion, skills and experience, to result in a valuable asset which can be used to aid decision making. Knowledge may be explicit and/or tacit, individual and/or collective.

Since this is a knowledge-driven DSS, developing an Expert System requires performing Knowledge Management activities. Knowledge Management supports the process of transferring expertise from human experts to computers and back to humans, which is illustrated in Fig. 11.10 and thoroughly discussed in [20]. These activities are also referred as Knowledge Engineering [27]. Turban et al. identify four activities in the Knowledge Management process [26]:

- **Knowledge acquisition** (from experts or other sources)—activity required to explicit knowledge that eventually is still tacit or to combine knowledge which is already explicit;
- **Knowledge coding** (in the computer)—activity required to deal with abstract concepts (e.g., events, time, physical objects, beliefs) and their relations, which constitute the content of the knowledge base, involving a new scientific field called ontological engineering [21];
- **Knowledge inferencing**—activity required to deal with the reasoning capabilities that build higher-level knowledge from facts and rules using heuristics or other search approaches; and
- **Knowledge transfer**—activity required to deal with delivery of knowledge to the users, namely to non-experts, using adequate interfaces or environments.

SINGRAR design process encompassed all these activities, which will be briefly discussed in the following subsections.

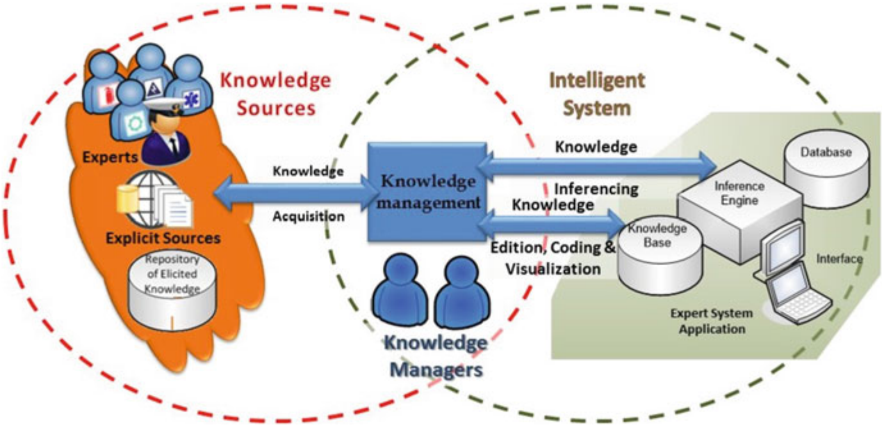


Fig. 11.10 Knowledge Management process used during SINGRAR design

11.3.1.1 Knowledge Acquisition

Knowledge acquisition involves gathering knowledge from different sources of tacit knowledge (i.e., unwritten knowledge residing in experts and decision-makers) and explicit knowledge (i.e., knowledge captured and coded in manuals, documents, drawings).

Tacit knowledge correspond to skills that are developed by subject matter experts as a result of years of experience, education and training. This knowledge is often difficult to be made explicit and is usually passed to others through demonstration and practice.

In the case of ships emergency management most decision-making processes involves tacit knowledge resulting from a mix of different forms of knowledge that are combined in the reasoning process used in the assessment of the situation and the judgement on the best course of action. The analysis of situation is usually supported by and integrates explicit knowledge, which is materialized by reference documents (e.g., doctrinal, organizational, procedural, technical).

The goal of Intelligent Systems is to emulate the human reasoning, therefore they need to embed experts' tacit knowledge which has to be captured (i.e., converted in explicit knowledge) and coded. This activity was critical in SINGRAR's design process, and involved the externalization of experts' tacit knowledge in terms of the ship's system engineering, warfare principles, technical procedures (e.g., damage control) and emergency management decision-making process, as well as embedding already explicit/documented knowledge, as illustrated in the left side of Fig. 11.10. Besides the compilation of already existing explicit knowledge, the elicitation of knowledge involved a long process of experts' interviews as well as the discussion of cases to identify the factors influencing the decision-making and their importance in different contexts. This process helped to enumerate and characterize the relevant Universes of Discourse required for the decision-making process, their relations and aggregation criteria.

11.3.1.2 Knowledge Coding

Knowledge coding is particularly challenging in what concerns recently externalized tacit knowledge, but can also be problematic for explicit knowledge, since such knowledge is to be either used to perform inference processes or as an output associated to the decision support processes.

Considering the vagueness of many of the concepts handled by SINGRAR, the approach used for knowledge coding was to represent such concepts by means of fuzzy sets, linguistic variables, and fuzzy relations.

In fact, the Fuzzy Set Theory is a generalization of Classical Set Theory that provides a way to incorporate the vagueness inherent to phenomena whose information is highly subjective and supplies a strict mathematical framework that allows its study with some precision and accuracy. Fuzzy Set Theory was formulated by Zadeh [29], in 1965, based on the principle that conventional quantitative techniques are not adequate to deal with humanistic systems (or similar complex systems). Humanistic systems are the ones that deal with problems using an approach comparable to human reasoning; while mechanistic systems reduce systems behavior to deterministic laws of mechanics, electromagnetism or thermodynamics.

The fuzziness treated by the Fuzzy Set Theory relates with the semantic interpretation of events, phenomena or statements, i.e., when there is some vagueness in the meaning of a concept. This fuzziness is present in most human activities, particularly the ones involving judgment, evaluation and decision based on natural language statements, since the meaning of the words is frequently vague and context dependent.

Fuzzy Set Theory provides a strict mathematical framework (arithmetic and logic) for studying conceptually vague phenomena in a quite precise and accurate way. This framework, which also encompasses the concept of linguistic variable, supports approximate reasoning and information extraction and processing in an increasing number of application domains (e.g., artificial intelligence, control engineering, decision theory, expert systems, logic, management, operations research, pattern recognition or robotics).

The basic concept is that a fuzzy set presents a boundary with a gradual contour, as illustrated in Fig. 11.11. This is the fundamental difference to classical sets, which present a discrete stepped border. In a classical set an element either belongs fully or is not member of a set (i.e., the membership to a classical set is either 1 or 0, meaning True or False in logical terms). The membership degree (μ) of an element to a fuzzy sets may be partial (i.e., its compatibility with the meaning of the set is limited), therefore intermediate degrees of truth are admitted ($\mu \in [0, 1]$). When $\mu = 0$ the element is not member of the set (i.e., is absolutely false that the element

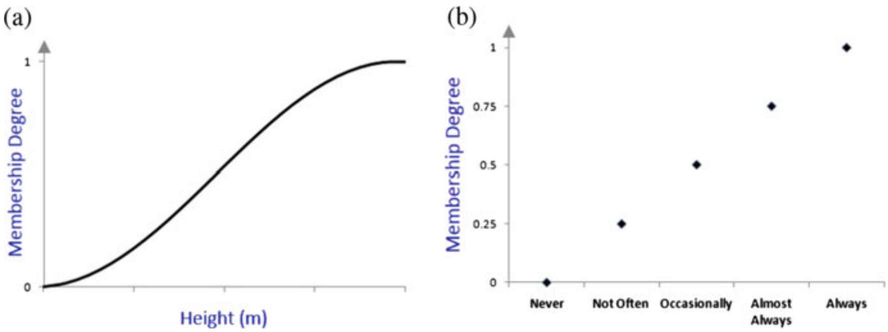


Fig. 11.11 Example of a continuous and a discrete fuzzy set. (**a**) Continuous fuzzy set. (**b**) Discrete fuzzy set

belongs to the set). The closer the value μ gets to 1 the bigger the affinity to the set. When $\mu = 1$ the element is a member of full right to the set (i.e., is absolutely true that the element belongs to the set).

Figure 11.11a presents an example of a continuous fuzzy set, representing the concept of "tall man", while Fig. 11.11b illustrates a discrete fuzzy set, in this case a linguistic variable related with the concept of "periodicity".

A linguistic variable is a variable whose values are words or sentences in a natural or artificial language. In technical terms, a linguistic variable is characterized by a quintuple which combines the name of the variable, a term-set (i.e., a collection of linguistic values), a universe of discourse, a syntactic rule (which generates the terms in the term-set), and a semantic rule (that associates with each linguistic value its meaning, which denotes a fuzzy subset of universe of discourse) [30–32].

Fuzzy sets admit a set of basic operations such as union, intersection, complement, product, Cartesian product, concentration and dilation, which allow for the development of a fuzzy arithmetic and logic [29, 35]. These operations correspond to mathematical formulas.

The knowledge coding, which also involved settling an ontology for managing shipboard incidents, defined the contents of the knowledge base and of the working memory, since both are compatible. SINGRAR knowledge base and working memory were implemented in a relational database whose data model is shown in Fig. 11.12 (the labels of the entities and relations are in Portuguese). A substantial part of the variables and relations contained in the data model is presented and discussed in Sect. 12.3 of Chap. 12 where the problem solving methodology is described.

11.3.1.3 Knowledge Inferencing

The approach used for knowledge inferencing in SINGRAR is based on fuzzy logic [33].

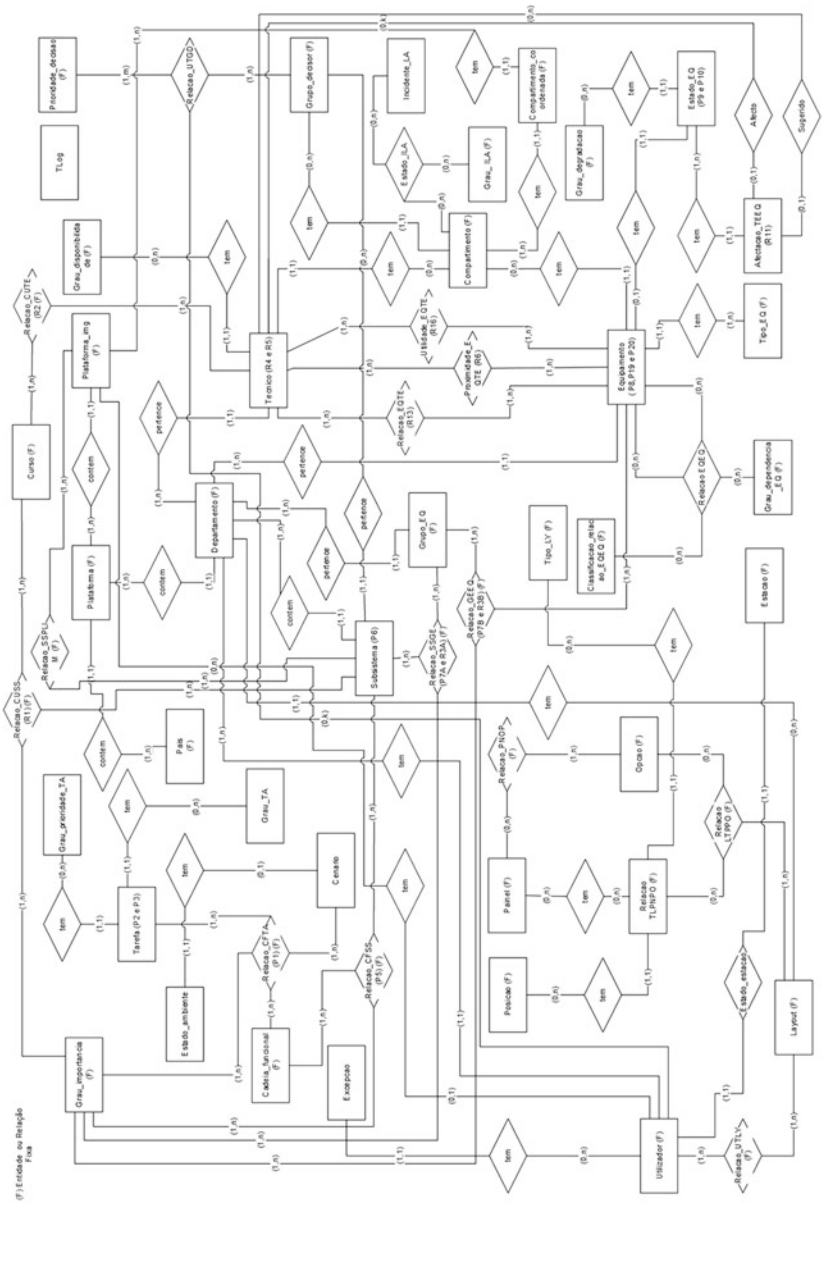


Fig. 11.12 Data model used to represent the database structure combining Knowledge Base and Working Memory components of SINGRAR

Zadeh [34] noting that fuzzy logic is not fuzzy, argues that this is a precise logic of imprecision and approximate reasoning, stating that:

> fuzzy logic may be viewed as an attempt at formalization/mechanization of two remarkable human capabilities. First, the capability to converse, reason and make rational decisions in an environment of imprecision, uncertainty, incompleteness of information, conflicting information, partiality of truth and partiality of possibility - in short, in an environment of imperfect information. And second, the capability to perform a wide variety of physical and mental tasks without any measurements and any computations.

As illustrated in the right side of Fig. 11.10, knowledge inferencing deals with the reasoning capabilities that are imbedded in the expert system inference engine, which build higher-level knowledge from facts and rules.

The inference model was the research theme of the author's MSc thesis [22]. A thorough discussion regarding the inference model of SINGRAR is presented in Sect. 12.3 of Chap. 12.

11.3.1.4 Knowledge Transfer

A topic which is concurrent with knowledge transfer is the one of knowledge visualization, which is defined in [3] as:

> the use of complementary visual representations to transfer and create knowledge between at least two persons.

Naturally the transfer of knowledge may involve different types of media. While trying to mimic some human capabilities, intelligent systems tend to assume a humanistic role in the creation and transfer of knowledge to human users, where the visual dimension is key.

As Eppler and Pfister note, knowledge visualization exploits different ways of representing insights, experiences, gathered evidence, and know-how in order to share knowledge, create new knowledge, or apply knowledge to decision making (e.g., lists, trees, semantic networks, schemas translating the what/where/how/what/why), making use of the whole spectrum of graphic representations, ranging from simple hand-drawn sketches to immersive virtual 3D worlds [9]. Figure 11.13 depicts an example of a drawing which conveys very important knowledge about the impact on ship stability resulting from incidents that involve the flooding of specific

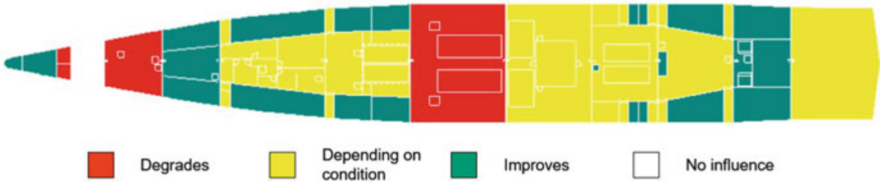


Fig. 11.13 Example of a graphical representation used in SINGRAR to share knowledge, in this case regarding the impact of floods in ship stability

compartments. The impact is color coded reflecting situations where the stability increases, decreases, is indifferent or depends on other factors.

SINGRAR is an expert system that deals with quite complex command and control problems. In the absence of support tools such situations are dealt with using procedures and recorded on boards using standard symbols that had to be incorporated in the representations used by SINGRAR.

Section 12.2 of Chap. 12 (User Interface Design) provides a very synthetic perspective of the SINGRAR user interfaces and offers some insights on the solutions found in this system to transfer knowledge to different users, according to their specific needs, roles and areas of responsibility.

11.3.2 Other SINGRAR Design Features

SINGRAR serves a community of users with a significant geographical distribution inside the ship, as it was illustrated in Fig. 11.7. Therefore, the system was designed considering a distributed architecture, using a Local Area Network (LAN) which links all the relevant compartments on board. This LAN supports the sharing of information and advice among all the workstations and offers opportunity for the implementation of other functionalities that improve the scope and the quality of the available services.

Some of such features are shown in Fig. 11.14. For instance, SINGRAR users benefit from a transparent access to external databases dedicated, for instance, to personnel and logistics management. The presence of personnel onboard is

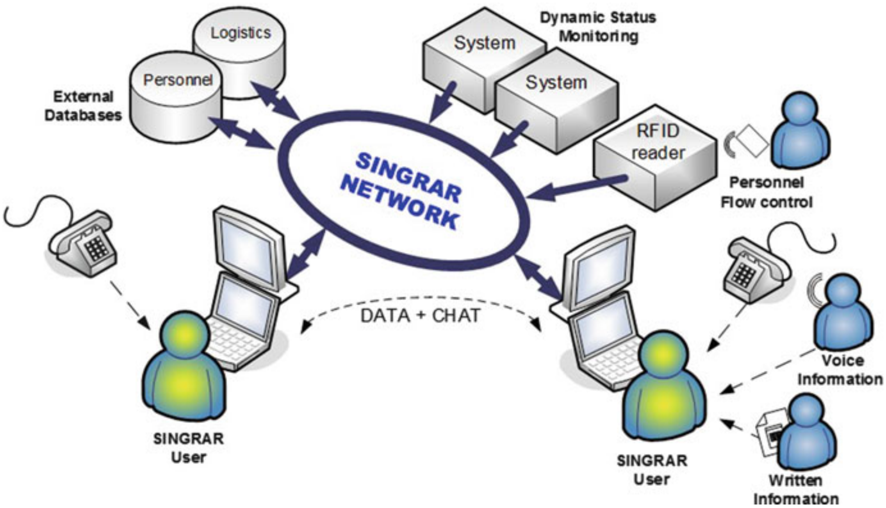


Fig. 11.14 Depiction of different sources of information used by SINGRAR and the flow of information sharing in SINGRAR LAN

controlled using an RFID-based system which feeds SINGRAR database to keep track on the human resources available. The status of the personnel, platform and systems is partially fed to SINGRAR by other existing systems, but a significant part of the situational facts are inserted manually as a result of the interaction of SINGRAR users with other crew elements. SINGRAR also provides functionalities for asynchronous messaging among users. These messages can be directed to specific users or broadcasted to the entire community.

11.4 System Development

SINGRAR development followed a spiral approach with loops of capabilities evolution, as illustrated in Fig. 11.15. Each evolutionary step resulted in the increase of functionalities, scope of decision-support and distribution, and also in the corresponding increase of the technical complexity of the solution.

This development process is discussed in the present section considering four dimensions:

- **System architecture**—addressing the issues related with system complexity levels and users' outreach;

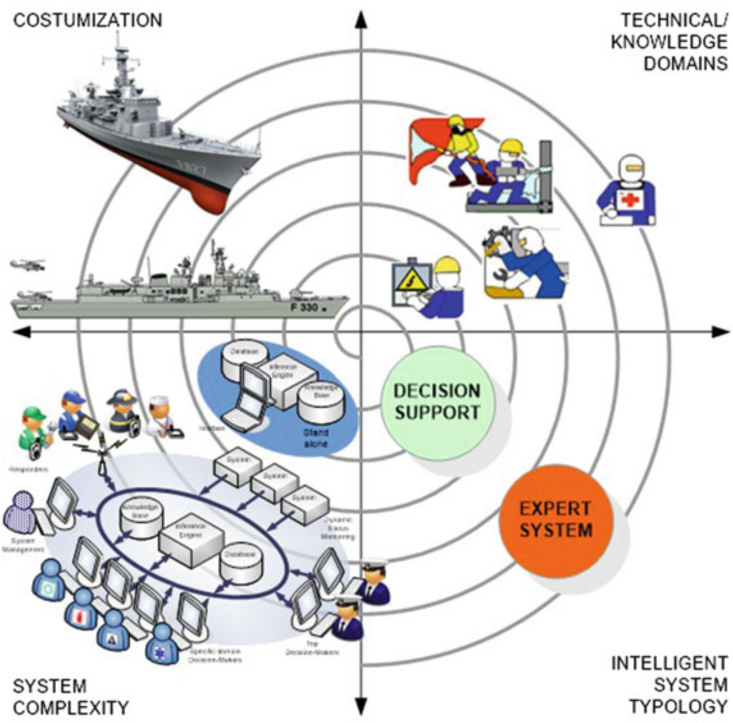


Fig. 11.15 SINGRAR spiral development

- **Intelligent System's typology**—addressing the issues related with the type of the decision-support provided;
- **Knowledge domains**—addressing the issues related with the scope of the knowledge encompassed by the system, namely in terms of technical expertise domains;
- **Customization**—addressing the issues related with the instantiation of the system to different application realities.

It will also be given a chronological perspective of the project development.

11.4.1 System Architecture

The project started about two decades ago with the approval by the Portuguese Navy Leadership of a proposal made by the author for the development of a concept for creation of a DSS, initially meant to support Weapon Engineering decision-makers. The feasibility of the concept and of the repair prioritization model was validated using a limited objective experiment which evaluated the results of a demonstrator (programmed in Microsoft Excel) that automated and improved the process described in Sect. 11.1.2 associated with the board presented in Fig. 11.2. The experiment was conducted in two phases: first, several sessions were conducted in parallel with table top exercises in which the decisions taken by decision-makers were confronted with demonstrator results and the discrepancies analyzed and discussed; second, the script of injects used during the Operational Sea Training was run and the results of the demonstrator compared against the expected decisions.

Following the validation of the concept a small development team was established, composed by the author (leading the team) and another Navy officer with education in computer science, both engaged with a workload of approximately 20 %, and one full-time programmer (also a Navy officer). The team initially focused on the development of a stand-alone DSS prototype for the Weapon Engineering Department. Since very early in the prototype development it was recognized the need to plan for expanding the scope of the decision support to other technical domains, and instead of creating a 'hard wired' structure it was taken the decision of developing the prototype as a customizable shell which supported the above described knowledge management activities and which could accommodate the flexible generation of interfaces of several typologies, whose contents would be edited and parametrized according to different ship characteristics in an edition/administration environment specially designed for this purpose thus accommodating virtually any type of ship. The prototype was developed for Microsoft Windows operating system using the Object Pascal language of Borland's Delphi IDE and the Corel Paradox relational database. The prototype was iteratively tested by prospective users and the solutions assessed using the user-centered design approach referred in Sect. 11.2.

Once the prototype reached a mature and validated state, the project evolved towards the development of a distributed solution. This new phase encompassed two different challenges: the implementation of a solution resilient to LAN failures; and the implementation of user profiles with differentiated permissions and working environments, and also other features to support communications and information sharing in a distributed environment (issue already addressed in Sect. 11.3.2).

Therefore, the main challenge was the implementation of a resilient structure which supported the required number of workstations considering battle or emergency situations. Figure 11.16 illustrates a typical distribution of SINGRAR workstations, manned when the ship activates Battle or Emergency Stations.

The three main decision centers (operations room, weapons engineering center, and machine control room) require a large number of workstations permanently manned that coordinate all emergency management activities. The damage control organization has three coordination cells also permanently manned. Several other workstations provide access for data input, status monitoring or action advice. These workstations, not permanently manned, are located for instance on equipment compartments, medical centers, and backup command centers. The initial configuration for the distributed system used a Client-Server architecture. However since this solution is not totally reliable and SINGRAR is most needed when it is more probable that equipment fail, namely the Servers and the LAN, a new approach to ensure survivability was pursued. Basically every workstation is able to operate independently from others; nevertheless the workstations actively look for others and try to cluster in a federation that can share data related with the situational status. If the data transfer infrastructure is working properly the behavior of the system is identical to a centralized system. In case of infrastructure failure SINGRAR performance degrades but the system is able to survive, with one or several groups of workstations sharing information. In the worst case every workstation operates in stand-alone mode and the information is updated manually. Implementing this approach presented some challenges, like the detection and management of the integration of newcomers to a group of workstations or the exclusion of 'missing' partners; or the fusion of data from different sources in order to provide a unique and coherent situational picture. Since these actions should be as transparent to users as possible, a multi-agent component was implemented and tested. The agents operate

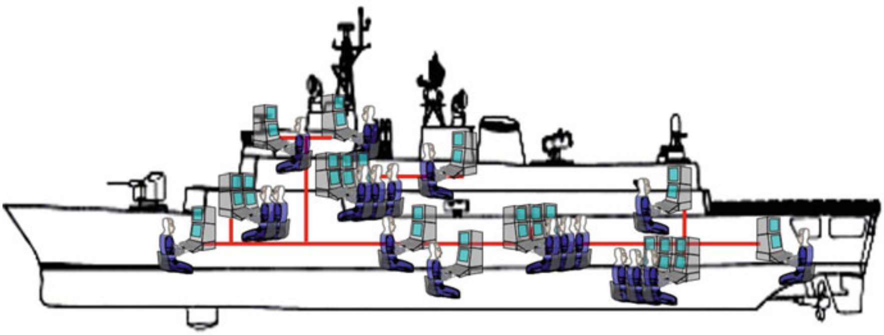


Fig. 11.16 Example of a typical distribution of SINGRAR workstations

autonomously monitoring and responding to changes in network partnerships, and synchronizing data contents on distributed database instances. Using this approach SINGRAR is able to operate with intermittent connections and still consolidate a common picture. The solution was implemented using a Microsoft SQL Server relational database.

11.4.2 Intelligent System's Typology

The type of decision-support provided by SINGRAR also evolved following the spiral approach. The first stage of development, concluded in 1999, resulted in the production of a fuzzy decision support system (FDSS) for emergency management. The model (described in Sect. 12.3 of Chap. 12) ensures the core functionalities of the system [22, 24], corresponding to the advice on repair priorities and resource assignment. The architecture of such FDSS was the one depicted in Fig. 11.9.

The subsequent phase of the project was the development of an expert system [23, 25] which further to advising on priority and resource assignment included new features expanding SINGRAR capabilities, namely adding explanations about the presented recommendations, and expert and contextualized support, namely for damage control operations. Some of the implications (in the knowledge inference model) of implementing this evolution are addressed in Sect. 12.3 of Chap. 12 ('Forward and backward chaining in the inference process').

11.4.3 Knowledge Domains

As it has been mentioned the scope of the knowledge encompassed by the system evolved from the initial concept up to the latest versions of SINGRAR, not only in terms of technical expertise domains but also in terms of support deepness.

In fact the initial goal was to deal with a very narrow scope problem domain. Despite already challenging, the first requirement was to provide support to Weapon Engineering decision-makers in setting repair priorities and managing the available technicians in a (quasi-)optimized way.

The knowledge acquisition process confirmed that limiting the problem domain to a single technical area would not be adequate, since weapon engineering equipment are client of critical services (e.g., power, cooling water, compressed air, ventilation) provided by other technical areas which had also to be considered in the decision process. Therefore, the natural evolution was to gradually aggregate the other technical domains responsible for ship's emergency management.

It became also obvious the benefits of using the same platform for managing damage control incidents (e.g., fires, floods). In fact, not only a significant number of decision-makers are common, but there is a permanent need for sharing information and using a common and consolidated picture about ship status. Thus, a new spiral of knowledge management activities was embraced to implement the required damage control capabilities in SINGRAR.

Both resource assignment and damage control require a strict monitoring of location, availability and status of the crew (e.g., current engagement and health status). Since personnel availability is contingent to health status and the report and control of casualties required also some support tool it became also natural to integrate this feature in SINGRAR. Furthermore, medical response has to consider all the type of path constraints that are discussed in Section 3.2 of Chap. 6 ('Resource assignment inference process'), therefore this internal organization uses SINGRAR to be aware of current hazards and to get advice.

In what regards support deepness, SINGRAR knowledge base contents grew providing contextualized access to procedural and technical documentation related with the incident that a particular user is dealing with in his/her interaction with SINGRAR, and the inference engine also explores in new ways the relations that are defined in the knowledge base, creating new knowledge. For instance, in support of troubleshooting activities a weapon technician can get access to maintenance manuals or to information about the services required by a specific equipment and their status; while a damage control coordinator can get access to the layout of-, hazards inside and in vicinity of- or firefighting equipment around a compartment where a fire ignited; or a decision-maker can get an impact analysis in terms of equipment operability and warfare capabilities resulting from shutting down a power load-center located in the vicinity of the compartment on fire.

The transition from a mono-expert to a multi-experts system was relatively smooth in what regards equipment repair prioritization and resource assignment, since the inference logics was basically the same, independent of the technical domain under consideration. Encompassing damage control was a quite different issue since the rational for dealing with this type of incidents follows a substantially different approach, requiring dedicated knowledge management activities. Nevertheless, the knowledge required for both areas complement each other and there are overlaps and communality, which resulted also in a synergistic effect of dealing with incidents from different standpoints.

11.4.4 Customization

The implementation of the first operational system ready to install onboard was a slow but steady process that took almost 10 years. This full scale system was customized to the "Vasco da Gama" class frigates. After a very successful testing and operational validation period initiated on 2004, all three ships of this class received the system and the necessary parametrizations to the specific context were performed. Since then incremental improvements were implemented, both in terms of features (as described above) and of usability (as discussed in Sect. 12.4 of Chap. 12).

In 2009, after performing a new knowledge management cycle (of approximately 2 years) conducted by the crew with the orientation and support of the development team, SINGRAR also entered in service onboard of the two “Bartolomeu Dias” class frigates (M-class frigates which were acquired to The Netherlands and up-graded). The parameterization of SINGRAR to this new context went quite smoothly and no major problems were faced.

11.4.5 SINGAR Project Chronology

As it became already obvious to readers, SINGRAR was nothing like a typical commercial engineering project, with well-defined initial requirements, a budget and a deadline. This project engaged very few human resources and the development counted with little funding (strictly the necessary to finance hardware acquisition and software licenses). Naturally this development model comes with a time cost. However, the time span was not necessarily bad since it made possible to mature concepts, assess alternative solutions and to benefit from the fast pace technological evolution that occurred meanwhile and which helped to solve some of the problems faced and to incorporate new or improved features.

Figure 11.17 presents a summary of the project chronology, since the initial concept, back in 1995, until the present.

A decisive moment in the project was its Operational Validation, in 2005, during the Operational Sea Training of the “Corte Real” frigate (the first one to receive SINGRAR onboard), performed in Plymouth (UK), at the Royal Navy’s Flag Officer Sea Training (FOST). The Assessment Report’s cover letter, signed by the Royal Navy Rear-Admiral Roger Ainsley reads:

> CORTE REAL has completed Portuguese Operational Sea Training to a Good standard [. . .]. This has been an outstanding achievement by a strongly led, capable Ship’s Company who should be rightly proud of this rare Good assessment, the first by a Portuguese ship at FOST and the first by a ship of any nation for over two years.

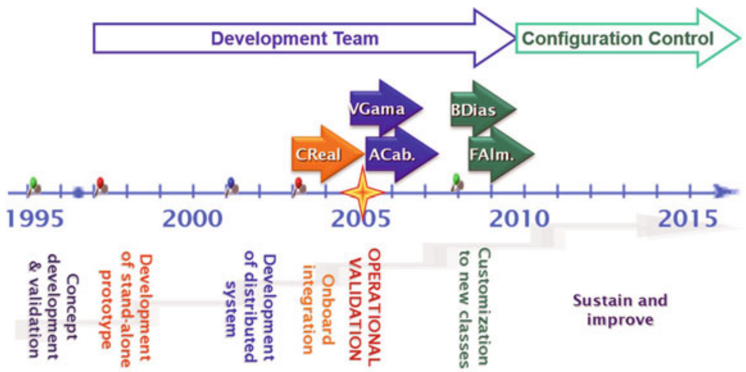


Fig. 11.17 Chronological perspective of the SINGRAR project

> [. . .] A culture of continuous improvement is also evident with innovative use of IT equipment in support of warfare. In particular, the SINGRAR system is supporting the Battle Damage Repair picture well and has exciting potential.

This was a relevant milestone for the recognition of SINGRAR as a valuable asset and a competitive advantage for the Portuguese Navy. Meanwhile, SINGRAR deserved the attention of allied nations interested in implementing the same type of intelligent support capabilities in their ships.

The cycle of integration of SINGRAR onboard of Portuguese Navy frigates was completed by 2010 and the project entered a new phase. The development team handed over to a configuration control team who took over the responsibility to sustain the system and to develop minor incremental evolutions.

Currently, the Portuguese Navy is committed to launch a project for the second generation of SINGRAR, which updates the technological platform used, and in articulation with the mid-life upgrade of the ships improves the integration of the SINGRAR and other Command and Control systems onboard, for information sharing, particularly regarding warfare information, platform status, and real-time monitoring of equipment health through the access to equipment built-in test data. The ultimate goal is that decision-makers, independently of their responsibility domain, take their decisions with full awareness of the warfare context and of the ship's capabilities and constraints, as illustrated in Fig. 11.18.

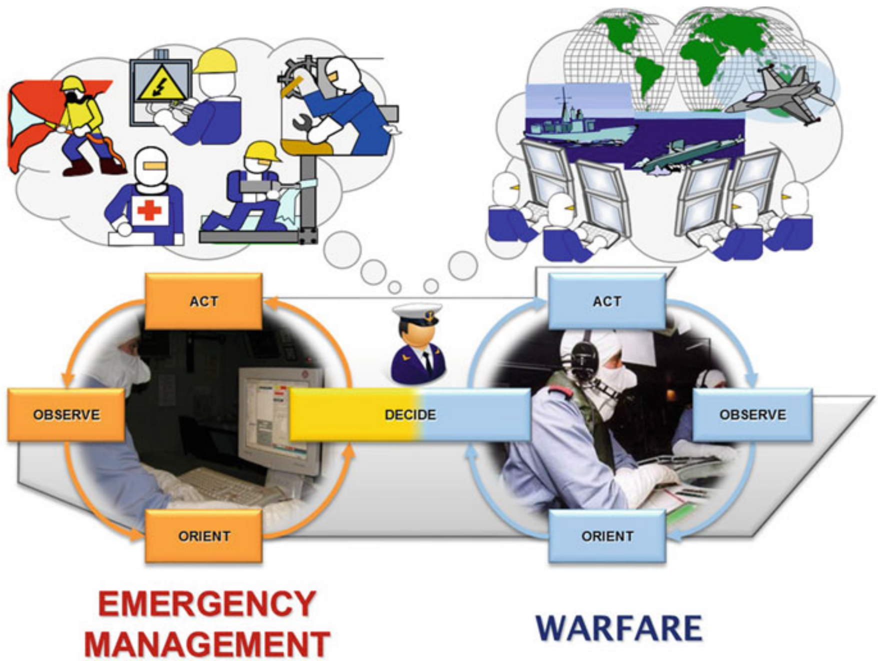


Fig. 11.18 Perspective of integrating information required for ship's decision-making processes

11.5 Conclusion

The present chapter addressed the conceptual part of SINGRAR case study. This part of the case study described the context of emergency management and the main features of SINGRAR, which was designed to be an information sharing platform that contributes to improve coordination, to ensure shared situational awareness, and to promote uniform and coherent recommendations regarding lines of action/resource assignment, contributing to the desired unity of effort of emergency responders.

SINGRAR spiral development allowed a steady evolution of the system in terms of architecture, intelligent system capabilities, and scope of knowledge domains addressed. The strategy of developing SINGRAR as a distributed expert system shell allowed not only to quite easily expand the scope and deepness of the support provided to different users inside a ship, but also to smoothly customize the system for use in a new class of ships.

Chapter 12 builds on this chapter presenting issues related with user interface design and system implementation, and offers insights on the usability study that was conducted. The conclusions presented also highlight the benefits of the distributed solution in support of collaborative emergency management decision-making.

References

1. Abou, S.C.: Fuzzy-logic-based network for complex systems risk assessment: application to ship performance analysis. Accid. Anal. Prev. **45**, 305–316 (2012)
2. Arendt, R., van Uden, E.: A decision-making module for aiding ship system automation design: a knowledge-based approach. Expert Syst. Appl. **38**(1), 410–416 (2011)
3. Burkhard, R.A.: Towards a framework and a model for knowledge visualization: synergies between information and knowledge visualization. In: Knowledge and Information Visualization: Searching for Synergies. Lecture Notes in Computer Science, vol. 3426, pp. 238–255. Springer, Berlin, Heidelberg (2005)
4. Calabrese, F., Corallo, A., Margherita, A., Zizzari, A.: A knowledge-based decision support system for shipboard damage control. Expert Syst. Appl. **39**(9), 8204–8211 (2012)
5. Cebi, S., Celik, M., Kahraman, C., Er, I.D.: An expert system towards solving ship auxiliary machinery troubleshooting: SHIPAMTSOLVER. Expert Syst. Appl. **36**(3), 7219–7227 (2009)
6. CEN: European Guide to good Practice in Knowledge Management - Part 1: Knowledge Management Framework. European Committee for Standardization, Brussels (2004)
7. Chauvin, C., Lardjane, S.: Decision making and strategies in an interaction situation: collision avoidance at sea. Transp. Res. Part F: Traff. Psychol. Behav. **11**(4), 259–269 (2008)
8. EMSA: Annual Overview of Maritime Casualties and Incidents 2015. European Maritime Safety Agency (2015)
9. Eppler, M.J., Pfister, R.: Best of both worlds: hybrid knowledge visualization in police crime fighting and military operations. In: Proceedings of the 13th International Conference on Knowledge Management and Knowledge Technologies, i-Know '13, New York, NY, 2013, pp. 17:1–17:8. ACM, New York

10. FEMA: Principles of Emergency Management. Federal Emergency Management Agency, Emmitsburg, MD, USA (2007)
11. Galton, A., Worboys, M.: An ontology of information for emergency management. In: 8th International Conference on Information Systems for Crisis Response and Management: From Early-Warning Systems to Preparedness and Training, ISCRAM 2011 (2011)
12. IMO: Resolution A.947(23) - Human Element Vision, Principles and Goals for the Organization. International Maritime Organization, London, UK (2004)
13. ISO: ISO 9241 Ergonomics of Human-System Interaction—Part 210: Human-Centred Design for Interactive Systems. International Organization for Standardization, Geneva, Switzerland (2010)
14. Kowalski, Z., Arendt, R., Meler-Kapcia, M., Zielski, S.: An expert system for aided design of ship systems automation. Expert Syst. Appl. **20**(3), 261–266 (2001)
15. Kowalski, Z., Meler-Kapcia, M., Zieliski, S., Drewka, M.: {CBR} methodology application in an expert system for aided design ship's engine room automation. Expert Syst. Appl. **29**(2), 256–263 (2005)
16. Lee, D.: Knowledge-based system for safety control of damaged ship. Knowl.-Based Syst. **19**(3), 187–191 (2006)
17. Lee, D., Lee, S.-S., Park, B.-J., Kim, S.-Y.: A study on the framework for survivability assessment system of damaged ships. Ocean Eng. **32**(89), 1122–1132 (2005)
18. Liu, Y., Yang, C., Du, X.: A multiagent-based simulation system for ship collision avoidance. In: Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues: Third International Conference on Intelligent Computing, ICIC 2007 Qingdao, August 21–24, 2007 Proceedings, pp. 316–326. Springer, Berlin, Heidelberg (2007)
19. Łozowicka, D.H.: Problems associated with evacuation from the ship in case of an emergency situation. Int. J. Autom. Comput. **3**(2), 165–168 (2006)
20. Nunes, I.L., Simões-Marques, M.: Contributions to the design of knowledge base development tools for intelligent systems. In: Ergonomics in Design: Methods and Techniques. CRC Press Taylor & Francis Group, Boca Raton, London, New York (2016)
21. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach, 3rd edn. Prentice Hall Press, Upper Saddle River, NJ (2010)
22. Simões-Marques, M.: Sistema de apoio à decisão difuso para a gestão de prioridades de reparação de equipamentos e afectação de recursos, em navios, em situação de combate (Fuzzy decision support system for equipment repair and resource assignment in ships under battle). Msc. thesis, Faculdade de Ciências e Tecnologia/Universidade Nova de Lisboa (1999)
23. Simões-Marques, M., Pires, F.J.: SINGRAR—a fuzzy distributed expert system to assist command and control activities in naval environment. Eur. J. Oper. Res. **145**(2), 343–362 (2003)
24. Simões-Marques, M., Ribeiro, R.A., Gameiro-Marques, A.: A fuzzy decision support system for equipment repair under battle conditions. Fuzzy Set. Syst. **115**(1), 141–157 (2000)
25. Simões-Marques, M., Nunes, L.I.: A fuzzy multicriteria methodology to manage priorities and resource assignment in critical situations. In: Humanitarian and Relief Logistics: Research Issues, Case Studies and Future Trends, pp. 129–153. Springer, New York, NY (2013)
26. Turban, E., Aronson, J.E., Liang, T.-P.: Decision Support Systems and Intelligent Systems, 7th edn. Prentice-Hall, Inc., Upper Saddle River, NJ (2004)
27. Turban, E., Sharda, R., Delen, D.: Decision Support and Business Intelligence Systems, 9th edn. Prentice Hall Press, Upper Saddle River, NJ (2010)
28. Vanem, E., Ellis, J.: Evaluating the cost-effectiveness of a monitoring system for improved evacuation from passenger ships. Safe. Sci. **48**(6), 788–802 (2010)
29. Zadeh, L.: Fuzzy sets. Inf. Control **8**(3), 338–353 (1965)
30. Zadeh, L.: The concept of a linguistic variable and its application to approximate reasoning-III. Inf. Sci. **9**(1), 43–80 (1975)
31. Zadeh, L.: The concept of a linguistic variable and its application to approximate reasoning I. Inform. Sci. **8**(3), 199–249 (1975)

32. Zadeh, L.: The concept of a linguistic variable and its application to approximate reasoning II. Inf. Sci. **8**(4), 301–357 (1975)
33. Zadeh, L.A.: Fuzzy logic = computing with words. IEEE Trans. Fuzzy Syst. **4**(2), 103–111 (1996)
34. Zadeh, L.A.: Is there a need for fuzzy logic? Inf. Sci. **178**(13), 2751–2779 (2008)
35. Zimmermann, H.-J.: Fuzzy Set Theory—and Its Applications. Springer, Dordrecht (2001)

Chapter 12
SINGRAR—A Distributed Expert System for Emergency Management: Implementation and Validation

Mário Simões-Marques

Abstract Complex Emergency Management operations require strong capabilities and robust decision-making support by means of Intelligent Systems. The chapter describes the main features implemented in the SINGRAR expert system to dynamically manage response priorities on emergency situations, based on situational parameters; to advice on resource assignment considering, for instance, individual capabilities, location and path constraints; and to offer expert support in specific technical domains, such as Damage Control. It also offers insights on the results of Usability study conducted to the system. SINGRAR was implemented in a customizable distributed shell that was developed to be scalable and adaptable to different scenarios. The implementation described addresses the management of critical incidents onboard of Navy ships.

12.1 Introduction

The current chapter describes the implementation and validation of SINGRAR. SINGRAR is the Portuguese acronym for Priority Management and Resource Assignment Integrated System. This system was developed to dynamically manage response priorities on critical situations based on situational parameters, and to advice on resource assignment considering, for instance, individual capabilities and location. According to the classification presented earlier in this book, by Power, SINGRAR can be deemed as a knowledge-driven DSS. The system was implemented in a customizable distributed shell that was developed to be scalable and adaptable to different scenarios. SINGRAR became operational a decade ago and addressed the management of critical incidents onboard of Navy ships, tackling equipment repair and damage control advice considering multithreat battle and emergency environments. Meanwhile the system was customized to different

M. Simões-Marques (✉)
Centro de Investigação Naval (CINAV) – Portuguese Navy, Alfeite, 2810-995 Almada, Portugal
e-mail: mj.simoes.marques@gmail.com

© Springer International Publishing Switzerland 2016

J. Papathanasiou et al. (eds.), *Real-World Decision Support Systems*,
Integrated Series in Information Systems 37,
DOI 10.1007/978-3-319-43916-7_12

classes of ships and received continuous improvements, both in terms of features and usability. Particularly regarding SINGRAR usability, an extensive study was conducted assessing its quality and identifying areas of improvement.

This chapter complements Chap. 11 in addressing the SINGRAR case study. In there it was described the context of emergency management and the main features of SINGRAR, a system designed to be an information sharing platform that contributes to improve coordination, to ensure shared situational awareness, and to promote uniform and coherent recommendations regarding lines of action/resource assignment, contributing to the desired unity of effort of emergency response.

The following sections will present the issues related with user interface design (Sect. 12.2) and system implementation (Sect. 12.3), and offer insights on a usability study that was conducted (Sect. 12.4). The final conclusions synthesize both parts of the case study highlighting the benefits of the distributed solution in support of collaborative emergency management decision-making.

12.2 User Interface Design

The context of use and the user requirements are determinant for the user interface design. As discussed in Sect. 11.2 of Chap. 11 (addressing SINGRAR's requirements analysis) there are different categories of SINGRAR operational users (from top level decision-makers to responders) acting on different specific domains (e.g., weapons, propulsion, power plant, damage control). Therefore SINGRAR has to offer user interfaces adapted to the roles and needs of this heterogeneous community.

SINGRAR interface design approach was guided by the principle that the system serves an organization structured on a hierarchy of emergency operations centers, where the scope of system interactions required by higher level decision-makers is broad and shallow, and scope of system interactions required by lower-level decision-makers and responders is narrow and deep. Naturally, as the scope broadens the greater is the complexity of the information presented to users and lesser the degree of detail that is possible to apprehend. Thus, a general coordination center is concerned with the overall picture, analyzing where the "hot spots" are and making a macroscopic management of resources, for example, moving available means to places where there is scarcity. On the other hand, a local decision-maker will be focused on responding to each individual incident, in real time, and making a discrete allocation of resources.

Nevertheless, the access to information should be flexible so that it can be used as needed, regardless of the level of the decision-maker. Thus, despite the quantity and quality of information presented to users is adjusted to the typical needs of their role, generically all information is available to all user levels. For example, a local decision-maker who struggles with limited resources can expand its field of view to understand what is happening in adjacent areas, allowing directed requests for assistance to those who have resources available.

Figures 12.1, 12.2, 12.3, and 12.4 illustrate examples of SINGRAR interfaces that combine different forms of conveying knowledge to users. In SINGRAR knowledge visualization assumes alternative graphical, tabular and textual formats which can be chosen by users according to their preferences to interact with the information that characterizes the operational environment and the status of the ship, the crew and the equipment, and also with the intelligent decision support provided by the system.

Figure 12.1 is the typical interface for top level decision-makers. The upper part of the screen (marked A) is a graphical presentation of the major incidents/anomalies affecting the ship and their equipment. A standard set of color coded icons provide the situational awareness about the ship systems' limitations and the severity of their impact considering the warfare context in which the ship is operating. The lower areas of the screen offers several contents which can be selected using "tab buttons". In the lower left side panel (marked B) it is possible to observe the list of repair priorities, also color coded, where it is added other relevant information, such as an estimation of time for the equipment to be back online. The lower right side panel (marked C) offers priority information regarding each specific technical area of the ship. Figure 12.1a presents a screen of SINGRAR customized for the "Vasco da Gama" class, while Fig. 12.1b shows the corresponding screen for the "Bartolomeu Dias" class. The screens were captured with years of difference and, despite some detail changes resulting from the evolution of user requirements, it is self-evident that a user which is familiar with SINGRAR in one class of ships can easily exploit SINGRAR in the other class of ships.

Figure 12.2 shows two screens dedicated to Damage Control decision-makers and responder coordinators. The situation depicted in this figure is the same shown in Fig. 12.1a. It is possible to see that the upper part of the screens of Figs. 12.1a and 12.2a coincide in the information that regards damage control incidents. The differences reflect information regarding other areas of responsibility (e.g., icons representing the status of weapons).

The lower part of Fig. 12.2a allows the access to information relative to the status and evolution of an incident and the visualization of contextualized knowledge regarding the local and the vicinity where it occurred (e.g., compartment layout, hazards, damage control equipment in vicinity), usually summarized in what is called a 'killcard'. Figure 12.2b shows a different interface that offers an expanded area to manage a particular incident, which also allows the access to contextualized knowledge. The particular screen presented in Fig. 12.2b illustrates a user interface for managing both an incident evolution and the resources engaged in controlling it (e.g., firefighters and their support teams).

Figure 12.3 offers an example of an interface designed for the management of operational context (threat warnings—marked A) and equipment status (marked B). Both screens show the same interface environment, which allows the interaction with several contents through the selection of different tab panels on the right side of the screen (marked C).

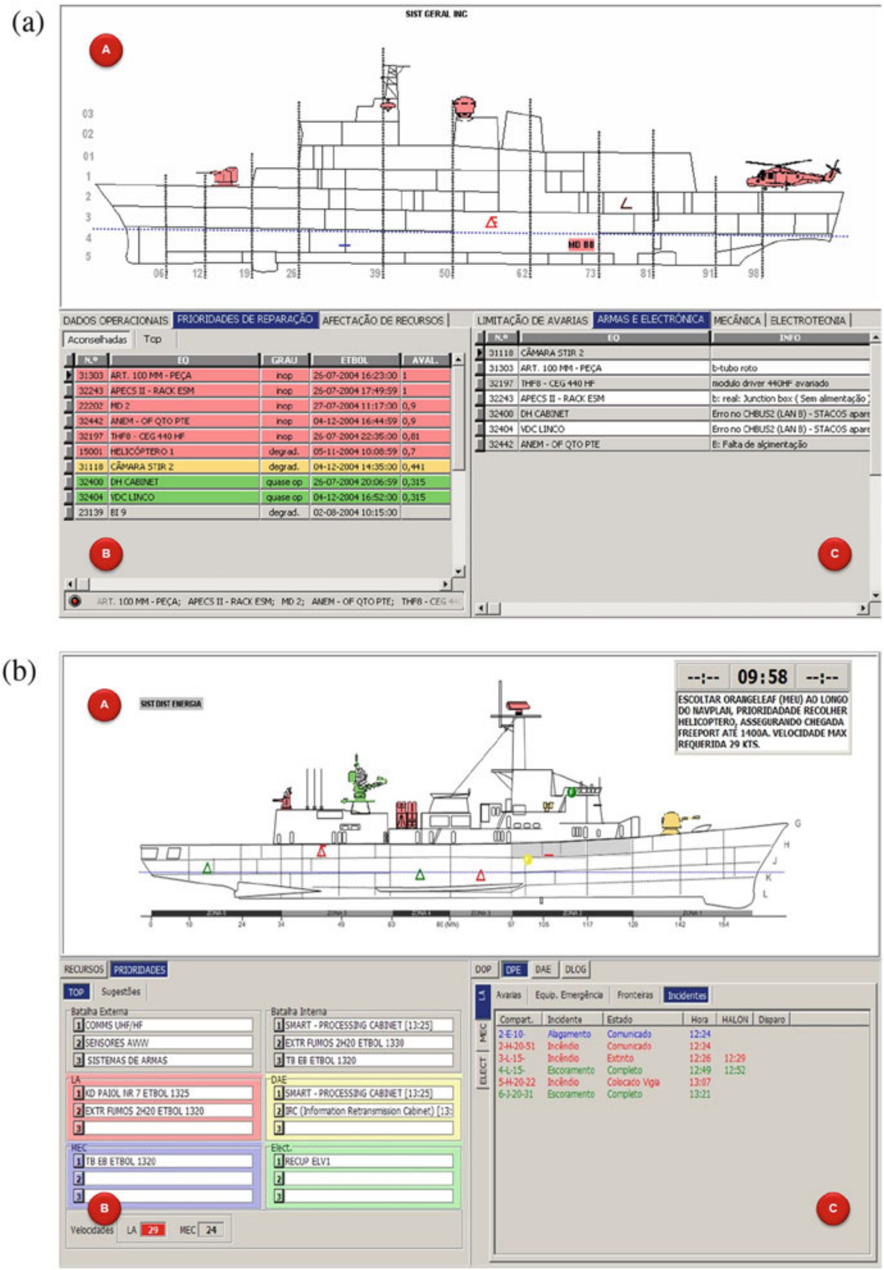


Fig. 12.1 SINGRAR interface designed for top level decision-makers, providing situational awareness (A), priority advice (B) and specific domain information (C). (**a**) "Vasco da Gama" Class. (**b**) "Bartolomeu Dias" Class

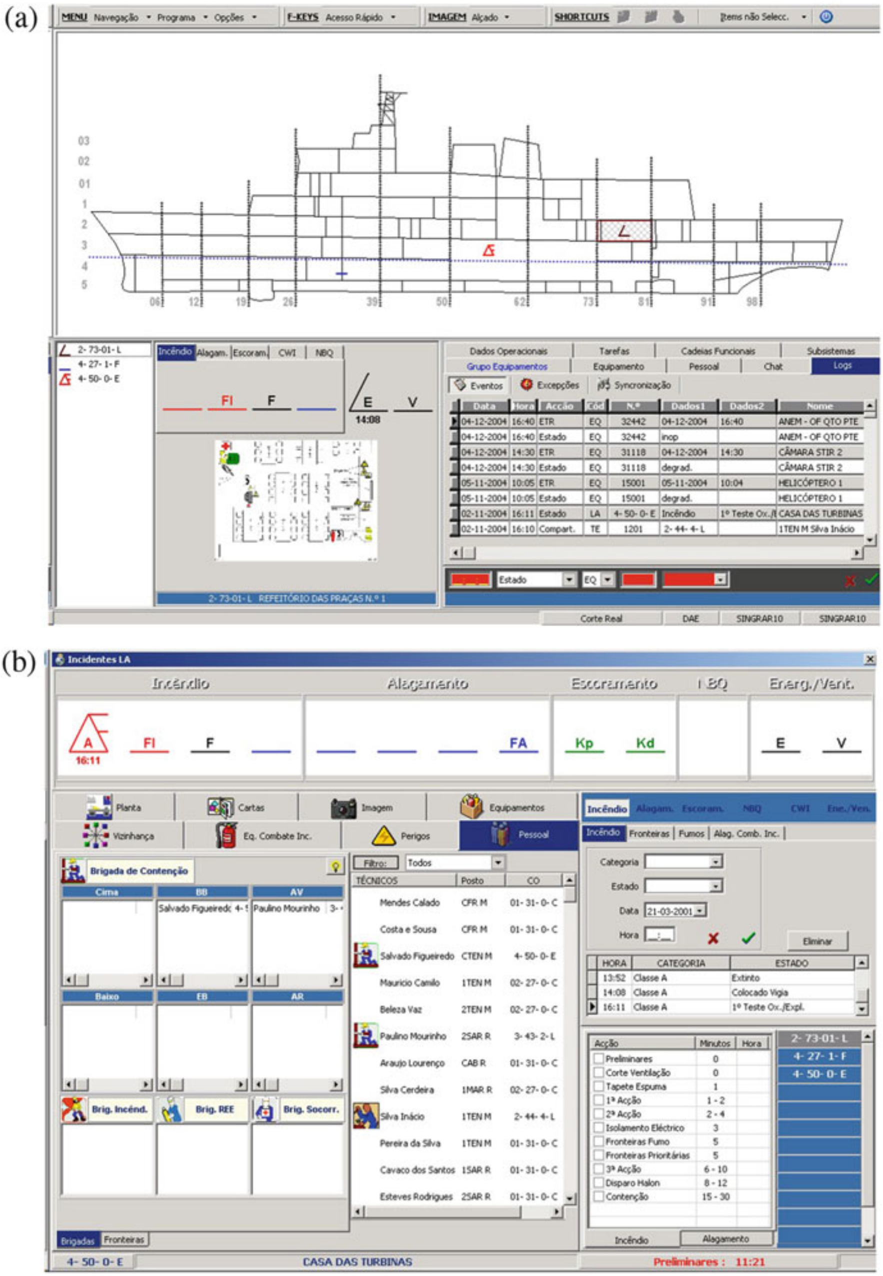


Fig. 12.2 SINGRAR interfaces to support damage control activities. (**a**) Incident management and access to compartment killcard information. (**b**) Incident and resource assignment management

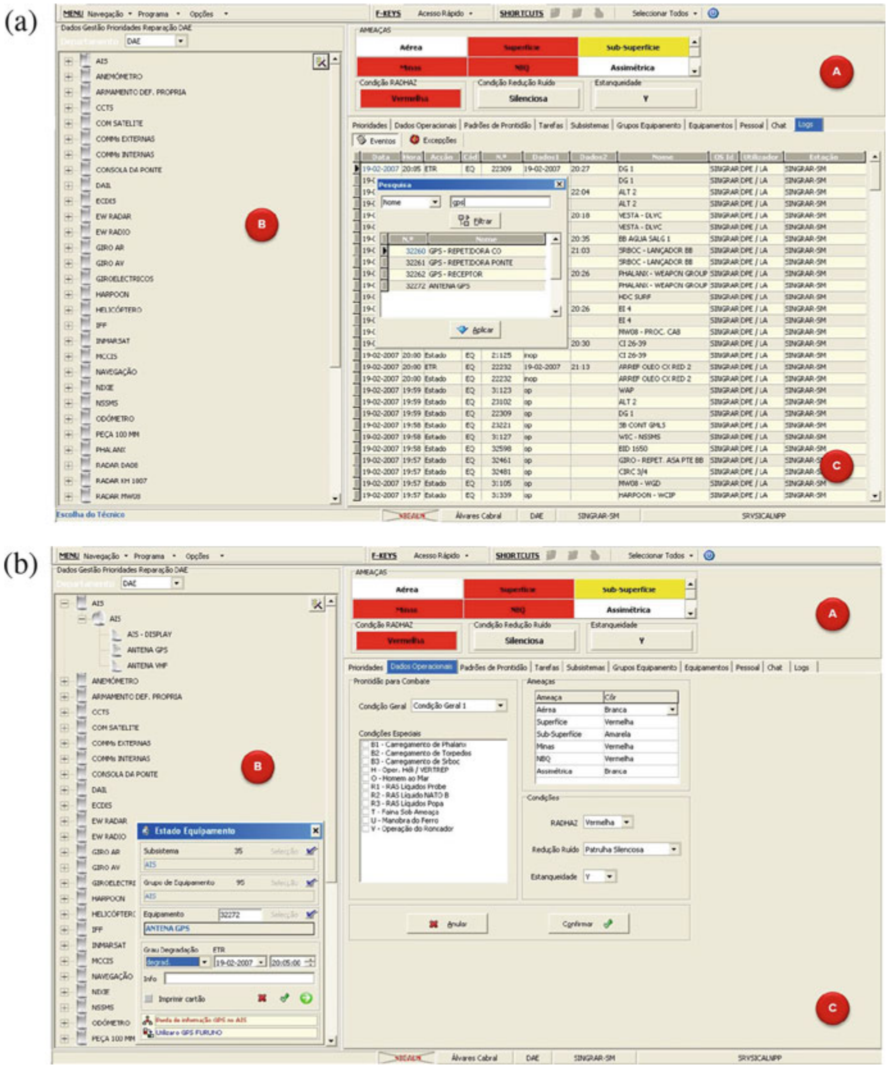


Fig. 12.3 Different tabs of the user interface, illustrated in pictures (a) and (b), offer access to complementary information and edition tools in the SINGRAR interface for the management of operational context (threats—A), equipment (status—B) and the access to additional parameters (C)

The example also illustrates one of the visual representations used in SINGRAR to access and edit equipment information; in this case equipment selection is done using the System-Equipment tree located on the left side of the interface (marked B). In fact, users can easily choose other forms of visual representation, as the one shown in Fig. 12.4, where each branch of the previous tree is presented graphically

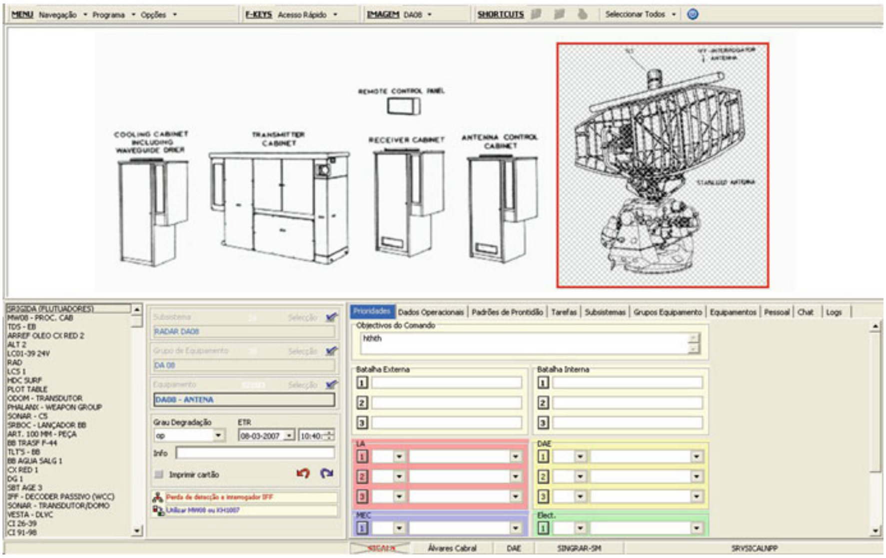


Fig. 12.4 SINGRAR provides alternative formats (e.g., graphical, tabular, tree) to access and edit information, accommodating different user preferences and needs

using an interactive representation of the equipment, which allows to access and edit the same information as the previous user interface.

Despite the complexity of the interfaces, the operation is relatively intuitive for users who are familiar with the tasks that the system supports. In fact, the system was designed to maximize compatibility with existing procedures and manual recording media (which is preserved as backup), allowing a quite smooth transition from one method to the other.

12.3 System Implementation

The complexity of the evaluation problem handled and the vagueness of most of the data under consideration led to the selection of an approximate reasoning approach for the development of the SINGRAR inference model, which conceptually can be interpreted as rule-based. In fact, the model was implemented using a Fuzzy Multiple Attribute Decision Making (FMADM) methodology [3, 20]. The next sub-sections discuss the main topics related with SINGRAR's knowledge inference processes.

12.3.1 Repair Priorities Inference Model

As mentioned before, from a conceptual point of view SINGRAR inference process can be interpreted as a chain of IF THEN rules.

Considering the equipment repair priority decision problem, a very high level rule can be adopted to characterize the reasoning process:

IF	equipment *i* utility is high	**AND**
	equipment *i* status is degraded	
THEN	equipment *i* repair priority is high	

where:

"*utility is high*"	is a fuzzy concept whose membership degree depends on importance of the task that equipment *i* performs, considering the current warfare scenario;
"*status is degraded*"	is a fuzzy concept whose membership degree depends on the level of operability of equipment *i*; and
"*repair priority is high*"	is a fuzzy concept whose membership degree conveys the notion of intervention urgency.

Such rule is the final one of an inference chain where, for instance, equipment utility is based on much more complex rules like the following:

IF	{ scenario 1 is active	**AND**
	utility of task 1 to scenario 1 is high	**AND**
	utility of system 1 to task 1 is high	**AND**
	utility of equipment *i* to system 1 is high}	
	OR	
	{...}	
	OR	
IF	{ scenario *m* is active	**AND**
	utility of task *n* to scenario *m* is high	**AND**
	utility of system *p* to task *n* is high	**AND**
	utility of equipment *i* to system *p* is high}	
THEN	equipment *i* utility is high	

Figure 12.5 offers a simplified perspective about the domains that are considered in the reasoning process (SC—warfare scenarios; TA—tasks; SY—systems; EQ—equipment). The cardinality of the domains shown in this figure just refers the universe of equipment handled by the Weapon Engineering Department of "Vasco da Gama" frigates. Obviously, due to the high number of equipment items onboard and to the high complexity of its interdependencies, defining Boolean rules for each relevant combination would be virtually impossible.

Therefore the approach used for SINGRAR is based on a fuzzy quantification of the degree of truth of each statement in the condition side of the rule, followed by its aggregation by means of a fuzzy intersection operator. For example, the conclusion of the first rule can be numerically computed using the following expression, where

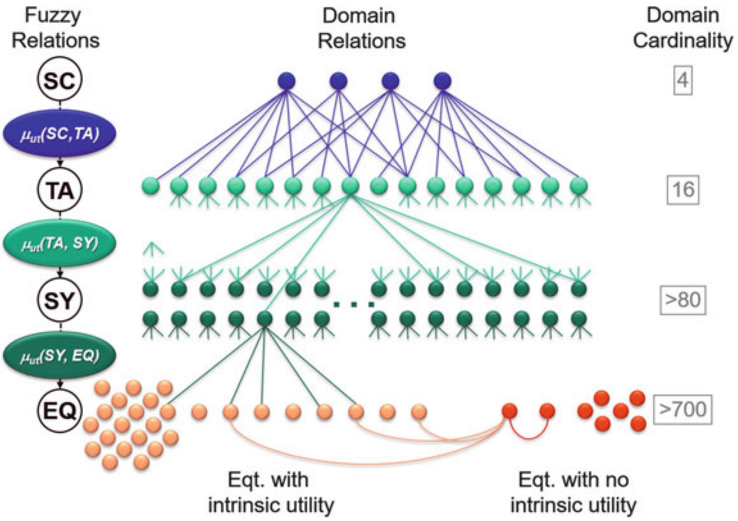


Fig. 12.5 Simplified characterization of the domains and fuzzy relations considered on prioritization reasoning process (the cardinality considers data regarding the Weapon Engineering Department of "Vasco da Gama" class frigates)

the numeric result is a measure of the degree of truth of the statement "*repair priority is high*":

$$\mu_{eqpr}(i) = \mu_{equt}(i) \otimes \mu_{eqst}(i)$$

where:

$\mu_{eqpr}(i)$	truth degree of the conclusion "*equipment i repair priority is high*"
$\mu_{equt}(i)$	truth degree of the condition "*equipment i utility is high*"
$\mu_{eqst}(i)$	truth degree of the condition "e*quipment i status is degraded*"
$\otimes$	fuzzy intersection operator (t-norm)

Both the condition statements and the conclusion are quantified in the interval [0, 1], where 0 means no priority, no utility or no degradation and 1 means the highest priority or utility, or total degradation. Intermediate values can represent different degrees of priority, utility or degradation. Identically, on the case of the second rule presented the assessment of the degree of truth of the statement "equipment utility is high" can be numerically computed, this time using the expression:

$$\mu_{equt}(i) = \bigcup_{i}\{\bigcap_{1,1,1}^{m,n,p} \mu_{utlevel_j}(i)\}$$

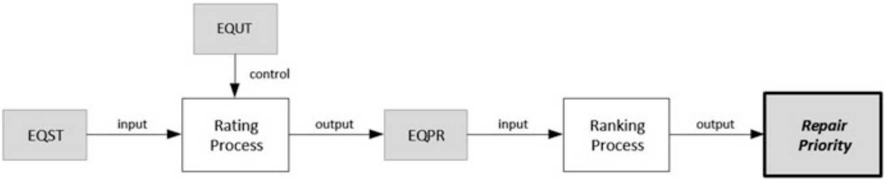


Fig. 12.6 Scheme of the SINGRAR repair priority component of the model

where:

$\mu_{equt}(i)$	truth degree of the conclusion "*equipment i utility is high*"
$\mu_{utlevel_j}$	truth degree of the *j*th level "utility" fuzzy relation
$\bigcup$	fuzzy union operator (t-conorm)
$\bigcap$	fuzzy intersection operator (t-norm)

Membership degrees can be obtained either by means of linguistic variables [17–19] or continuous membership functions. The relation between the antecedent part (IF) and the consequence part (THEN) of the rules is defined by means of the fuzzy relations, which are graphically illustrated in Fig. 12.5. Detailed descriptions of this part of the model and application examples can be found in [11–13].

Repair priority advice, illustrated in Fig. 12.6, is provided after the execution of a ranking process that sorts priority levels (EQPR) evaluated by the rating process, which corresponds to the execution of the IF-THEN rules given the status (EQST) and the utility (EQUT) of faulty equipment.

12.3.2 Resource Assignment Inference Model

An effective emergency management system requires the adoption of robust criteria for evaluating the adequacy of allocating a given resource to an incident response. Examples of factors to consider in assessing the suitability of the resources to assign to a specific incident are skills/capabilities, proximity, and availability. As for the repair priority process, the rational for assigning technicians to repairs can be expressed by IF THEN rules such as:

IF	equipment repair priority is high	**AND**
	technician utility to repair the equipment is high	
THEN	technician's assignment priority is high	

Identically, this rule is part of an inference chain where, for example, the evaluation of the technician utility to repair equipment is based on a new rule:

IF	technician skills to repair the equipment are high	**AND**
	technician availability is high	**AND**
	technician proximity to the equipment is high	
THEN	technician utility to repair the equipment is high	

The inference chain extends by further levels, involving the relations required to support the decision making process. Once again the implementation of the rules is based on fuzzy sets combined using fuzzy aggregation operators. Figure 12.7 presents as example the fuzzy set Proximity and illustrates the evaluation of the degrees of membership (μ_{T_1} and μ_{T_2}) for two technicians (T_1 and T_2) considering their respective distance (d_{T_1} and d_{T_2}) to equipment EQ.

However, solving the resource assignment problem realistically in the context of battle or emergency presents further challenges. This subsection will briefly address a new method developed for SINGRAR to identify the shortest unconstrained (or less constrained) path, to assess equipment to technician proximity. Note that for reasons related both with personnel survivability and ship's coverage the crew is distributed by the compartments of the ship, which in a frigate exceed 400.

For dealing with the problem, one as to recall that ships are a particular type of construction that (for safety reasons) impose constraints in connection of adjacent spaces, affecting the options of possible paths linking the compartments. Dramatic accidents, like the sinking of Titanic, taught invaluable lessons that determined ships' architecture; for instance, dividing their length in several watertight sections improves ship survivability in the presence of major damage. This architecture may

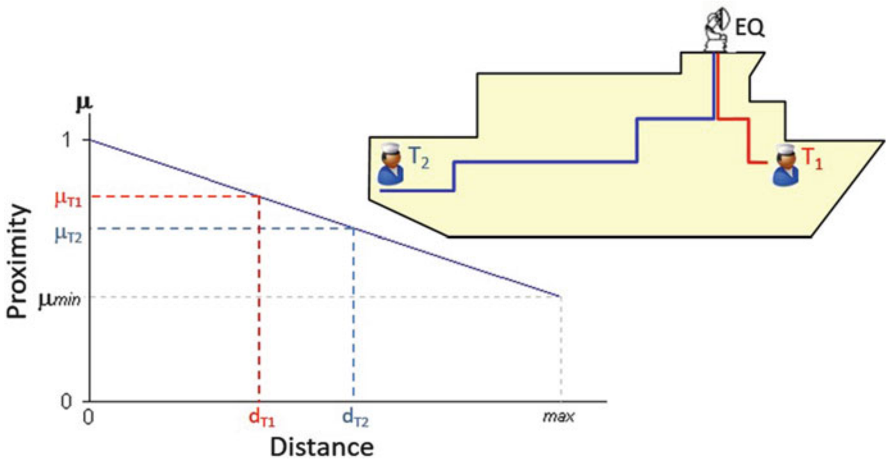


Fig. 12.7 Evaluation of the Proximity fuzzy set membership degrees (μ_{T_1} and μ_{T_2}) considering the distances (d_{T_1} and d_{T_2}) between equipment EQ and technicians T_1 and T_2

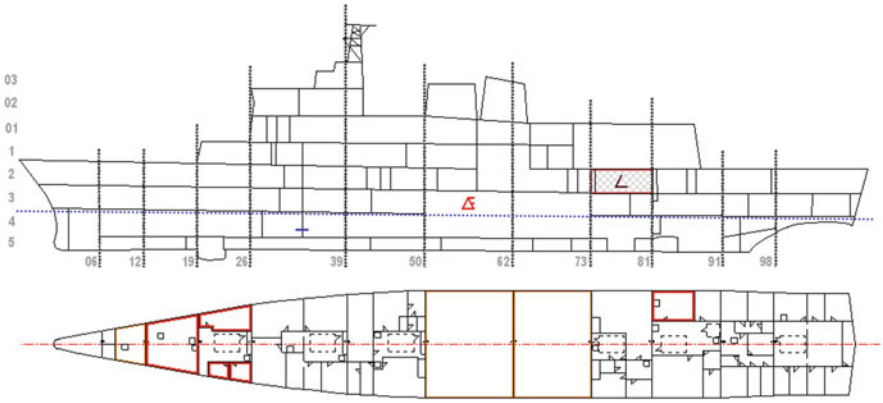


Fig. 12.8 Illustration of the internal structure of a ship

force a sailor in a compartment below the water line to go up and down several decks when headed to an adjacent compartment on the other side of a watertight bulkhead.

Figure 12.8 illustrates two views of the plans of a "Vasco da Gama" frigate. The dotted vertical lines in the upper drawing mark the separation of the sections in which the ship is divided. The lower drawing represents deck 3, located at water line level, where one can note that there are no doors in the bulkheads corresponding to such separations.

Accounting for this type of peculiarities in the definition of paths between compartments is necessary, but not really a big issue, since a set of simple heuristic rules can be used. However, when particular conditions impose additional transit constraints, offering advice on the paths to follow and on specific requirements to observe becomes a more complicated task. For instance, the shadowed area in the top drawing of Fig. 12.8 (at Deck 2, between bulks 73 and 81) presents an open triangle symbol which signals a fire that is being fought. Naturally, besides the firefighting team, no one else is supposed to stay in the surrounding area. This may be very problematic for the transit of other emergency teams, since Deck 2 is a "highway" for circulating inside the ship. If the fire is small and exceptional situations force personnel to pass in that area, special requirements have to be observed not only in terms of control of the presence of such individuals but, particularly, regarding the need for wearing personal protective equipment (PPE) (for instance, autonomous breathing apparatus).

For a ship on operations the constraints of crew circulation are frequent even when no accidents have occurred. It is easy to enumerate examples of operational conditions that restrict the crew to freely access some areas, including the technicians that have to go troubleshoot and repair equipment. For instance, the hangar and the flight deck are areas restricted to air crews and to aircraft support team; no one else is supposed to go there without a reason and without control. Several outside decks have weapon systems that may move or fire without prior warning; once again no one is supposed to stay or pass in these areas; when exceptional

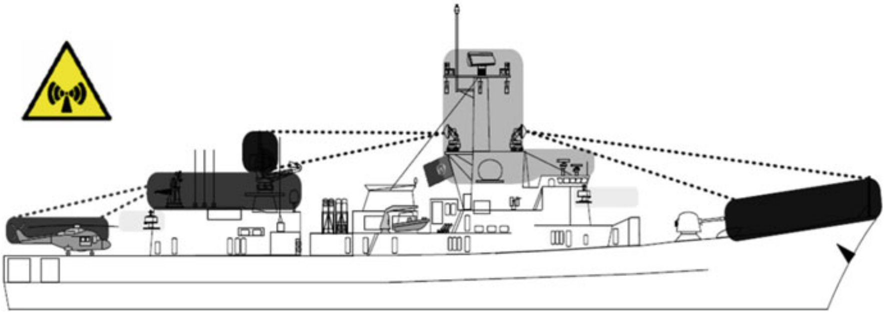


Fig. 12.9 Example of the areas affected by constraints due to armful electromagnetic radiations

situations require the presence of personnel, weapon operators must be aware of their presence and, if possible, inhibit the motion and firing of such systems. Sensors (e.g., radars), communication systems and some weapons emit high power electromagnetic radiations, which are dangerous for maintainers that have to access equipment on masts or even for people that pass in their neighborhood; once again extra control and actions are required when personnel has to access to or travel in such areas (Fig. 12.9 illustrates this situation). The examples could go on, but the point is made—it is quite likely for a warship in operations to have situations affecting the freedom of circulation using the shortest-path. When such constraints arise (due to normal operation of systems or because of accidents), SINGRAR is supposed to have information about them, assess alternative shortest-paths, identify if there are any unconstrained paths and, if not, provide advice on the lines of action required (e.g., use of PPE, who to inform, activation of safety procedures). As mentioned before, this information is also used for the prioritization/selection of alternative technicians to assign to the repair of specific equipment. Thus, selecting the technician which is closer to such equipment may not necessarily be the right choice, if the cost of fulfilling all the requirements necessary to overcome the existing constraints is very high.

For solving this problem the ship can be modeled as a connected network. Considering that each compartment and relevant space of a ship is a node, a network for representing a “Vasco da Gama” frigate is composed by about 400 nodes since, as mentioned before, this is the approximate number of compartments. In practice just a small portion of the full set of nodes integrates the paths that connect pairs of source-destinations nodes, since most of the compartments are dead ends (i.e., many nodes present only one arc). For reducing the complexity of decision-making activities based on such network it would be convenient to change the granularity of the problem and represent it in a meta-network,[1] i.e. a network-of-networks

[1]Carley [1] discusses meta-network in the context of organizations and defines a meta-network as the set of personnel, resources, knowledge, tasks and the relations among those entities. In this context Carley mentions that a meta-network is comprised of a variety of sub-networks including,

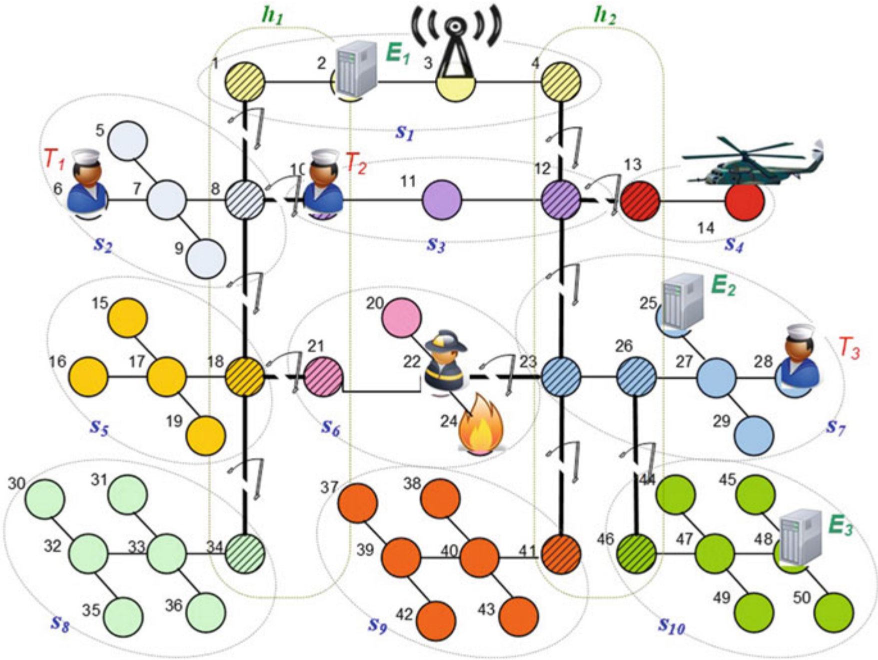


Fig. 12.10 Example of the representation of a ship internal structure as a network

(concepts addressed, for instance, by Panzarasa et al. [10], Carley [1], Carley et al. [2], and Joslyn [5]).

Onboard of a ship it is easy to recognize the existence of a meta-network composed by sets of compartments, equipment, personnel, etc. In fact this is implicit in the SINGRAR model. For the purpose of the present chapter the meta-network analysis will be focused on the physical structure of the ship and on the organization of the compartments.

Thus, considering the ships' architecture it is possible to define a meta-network composed by a number of sub-networks that connect with others in single nodes. The definition of the sub-networks becomes natural due to the structure of the ship's decks, bulkheads and passage hatches. Figure 12.10 provides a manageable example that will be used to illustrate the approach adopted.

The full network is composed by 50 nodes (numbered 1–50). Considering the layout of the network ten sub-networks were identified (which are color coded and designated s_1 to s_{10}) composed by sets of nodes (e.g., sub-network s_1 groups nodes 1–4) that are separated from the others by hatches, illustrated by the symbol over the connecting arc (see for instance the arc connecting nodes 1 and 8). A second type

but not limited to, the social network (interactions among personnel), the knowledge network (who knows what), and the precedence network (what tasks come before what).

of sub-networks, designated as hubs, is considered. Hubs are composed by nodes that connect latter sub-networks (the nodes drawn with a fill pattern). There are two hubs in Fig. 12.10, identified as h_1 and h_2.

In the example of Fig. 12.10 three sub-networks present constraints that affect the circulation or the presence of crew in the area: (1) sub-network s_1 has a high power emitter in node 3 which presents a radiation hazard; (2) sub-network s_4 is an area reserved for flight operations; and (3) sub-network s_6 is affected by a fire in node 24, and there are firefighting operations in the area.

The problem to solve is to assign technicians to the repair of equipment. In this case there are three technicians (T_1 to T_3) placed at nodes 6, 10 and 28; and three equipment (E_1 to E_3) located at nodes 2, 25 and 48. For the sake of clarity of the explanation, some simplifying assumptions are taken. For instance, it is assumed that the technicians are all equally skilled and available to perform the repairs. On the other hand it is assumed that all equipment have the same repair priority. Therefore, the only attribute to consider is the distance separating technicians from equipment and the constraints affecting the circulation.

The first step to consider is to convert the network into a meta-network that changes the granularity of the problem. Figure 12.11 illustrates one possible approach, based on the previous structure identified. Each of the nodes corresponds to a sub-network (s_1 to s_{10}) or a hub (h_1 to h_2). The connections of sub-networks are done through hubs; therefore an arc always connects a sub-network node to a hub node. The location of the constraints and the sources and destinations of the paths are also shown in the meta-network.

The second step is to identify the paths in the meta-network that connect sources to destinations, in order to define the sets of nodes required to use by the shortest-path solving algorithm.

This procedure starts by identifying pairs of sources and destinations that are located in the same sub-network. In this case there is only one such case, which corresponds to the pair $T_3 - E_2$, located in s_7. If there are no conflicts (such as

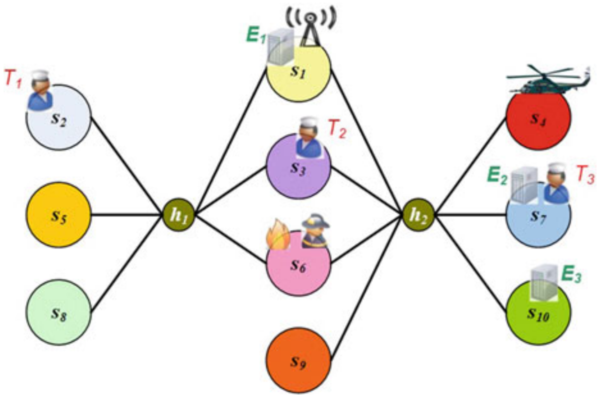


Fig. 12.11 Change of problem granularity using a meta-network

more than one technician or more than one equipment in the same node of the meta-network) the assignment may be done immediately avoiding further data analysis. If not, the procedure can include the evaluation of distances for assigning the existing technicians to repairs. If there are extra technicians or equipment to repair they will be considered together with the others present in the meta-network. In the case where the assignee and the task coexist in the same location the local constraints are not much relevant for the decision. If there are constraints they may be minor, not really affecting the repair (e.g., it may be mainly a matter of authorization and control of access to sensitive areas); on other situations the constraints may affect performing the work (e.g., need for special protective equipment) not being really impeditive of its execution; finally some constraints may be the cause of the fault or may impede the work of being performed (e.g., a fire may damage equipment and preclude the repair of materiel in the surroundings). Anyway if a technician can stay in the area while a constraint is active, it is assumed that the requirements for dealing with the constraint were already observed.

After all local assignments have been done, the procedure proceeds with the analysis of the remaining situations. In this case technicians T_1 and T_2 need to be paired to equipment E_1 and E_3. The first action corresponds to identify if the meta-network presents any unconstrained paths connecting the source and the destination nodes. In this example, it is possible to find unconstrained paths for accessing s_{10} for both technicians available. The unconstrained path for T_1 towards E_3 is $s_2 - h_1 - s_3 - h_2 - s_{10}$. Therefore assessing the shortest-path is done based a network composed by the nodes resulting from the union of these sub-networks (see Fig. 12.12a). The unconstrained path for T_2 towards E_3 is $s_3 - h_2 - s_{10}$. Identically the distance evaluation uses the network resulting from the union of these sub-networks (see Fig. 12.12b). Figure 12.12 illustrates the resulting networks. Assuming that the arcs length is unitary it is possible to verify that the distance for the pair $T_1 - E_3$ is 10, while distance for the pair $T_2 - E_3$ is 7. Naturally this distance depends on the "cost"

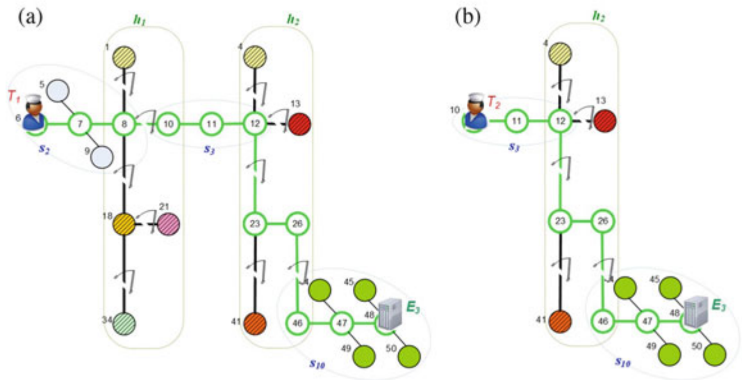


Fig. 12.12 Networks used for assessing the shortest-path connecting technician-equipment pairs. **(a)** $T_1 - E_3$. **(b)** $T_2 - E_3$

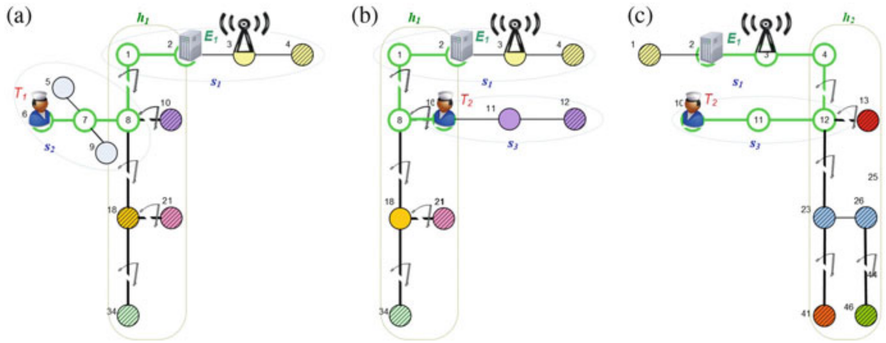


Fig. 12.13 Networks used for assessing the shortest-path connecting technician-equipment pairs. (**a**) $T_1 - E_1$. (**b**) $T_2 - E_1$. (**c**) $T_2 - E_1$

associated with crossing a hatch which in this case is assumed to be also unitary (for instance if an arc with a hatch is considered five times harder to cross than the other arcs, then the distances would be: 22 for pair $T_1 - E_3$; and 15 for pair $T_1 - E_3$.)

Note that in these cases, where the pairing is done for assignees/jobs located in opposite extremes of the network, the reduction in the number of nodes used in the shortest-path solving problem was about 50 and 70 %, depending on the pair considered, since the initial network has 50 nodes and, respectively, only 24 or 15 had to be passed from the database and used by the adopted shortest-path solving algorithm.

The final procedure, within step 2, has to deal with the remaining cases where no unconstrained paths were found. In this case, the repair of E_1 will always happen in a constrained sub-network (s_1). In this case the procedure has to evaluate the distance of the possible pairs, and to identify the type of constraint. Since the assignment of technicians T_1 and T_2 was not yet decided, the possible pairs are $T_1 - E_1$ and $T_2 - E_3$. Returning to the meta-network (Fig. 12.11) it is possible to identify that the constrained path connecting T_1 and E_1 is $s_2 - h_1 - s_1$ and two constrained paths connecting T_2 to E_1 the first being $s_3 - h_1 - s_1$ and the second $s_3 - h_2 - s_1$. Therefore assessing the shortest-path is done based a network composed by the nodes resulting from the union of these sub-networks (see Fig. 12.13a–c). Considering arcs of unitary length the distance for the pair $T_1 - E_1$ is 4, while the minimum distance for the pair $T_2 - E_1$ is 3 (corresponding to Fig. 12.13b). However, in this case, besides the distance there is a second parameter to consider, which is the nature and impact of the existing constraint (a Radiation Hazard).

In these cases, the pairing is done for assignees/jobs located quite close in the network, therefore the reduction in the number of nodes used in the shortest-path solving problem was at least 74 %, for each pair, since from the initial 50 nodes only 13, 12 or 11 (depending on the sub-networks/hubs involved) had to be passed from the database and used by the adopted shortest-path solving algorithm.

The final step is to apply the resource assignment method and criteria. Figure 12.14 represents the network representing the analyzed assignment problem.

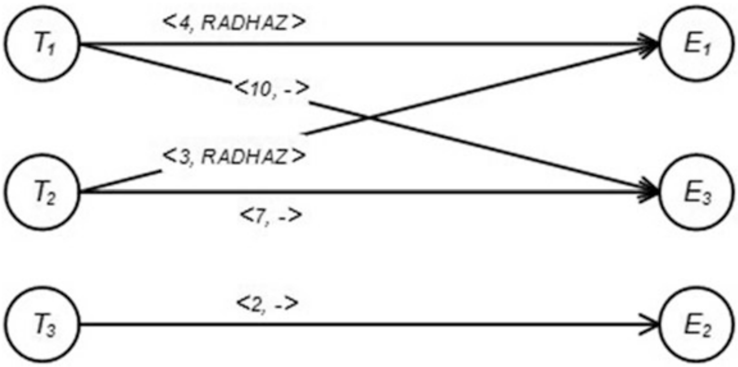


Fig. 12.14 Network representing the analyzed assignment problem

Considering that the goal is to minimize the combined cost resulting from distance and constraints, the optimal solution for the problem is assigning T_1 to E_1, T_2 to E_3 and T_3 to E_2, with an accumulated distance of 13, and the need for fulfilling the requirements related with T_1 going to an area affected by a Radiation Hazard (RADHAZ).

The resource assignment procedure described can be generalized and structured in a five phases metaheuristic approach:

Phase 0 **Pre-processing**, where the complex network is used to generate a meta-network that reduces the complexity of the original network by representing sub-networks and hubs as single nodes. This is done prior to the analysis;

Phase 1 **Data Collection**, namely the listing of available assignees (including location), pending tasks (including location) and existing constraints (location and type);

Phase 2 **Assessing Distances and Constraints**, this is the core algorithm and comprehends the following steps:

1. Find intra sub-network paths;
2. Find unconstrained shortest-paths;
3. Find constrained shortest-paths;

Phase 3 **Resource Assignment**, based in the results of the previous phase use repeatedly a multiple-objective resource assignment solving method to complete the Resource assignment list, as follows:

- Run a multiple-objective resource assignment algorithm on the Pending resource assignment list
- Add ‘assignee – task – constraints – path’ tuples to Resource assignment list
- Remove assigned assignees from the list of available assignees
- Remove assigned tasks from the list of pending tasks

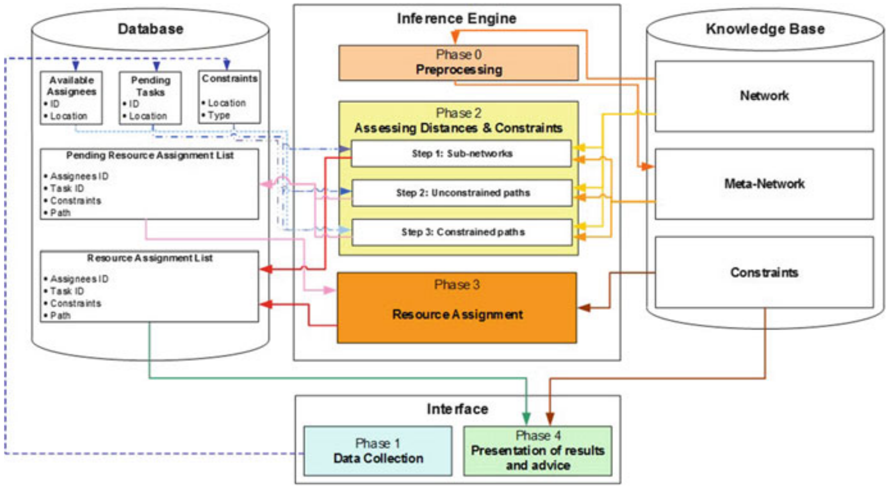


Fig. 12.15 The resource assignment metaheuristic shown in the Expert System context

Phase 4 **Presentation and Explanation of results and Advice on Constraint mitigation actions**, present resource assignment advice and offer explanations about the underlying reasoning process; offers additional types of advice, particularly regarding actions required to deal with existing constraints.

This metaheuristic is illustrated in Fig. 12.15 in the context of the Expert System architecture, placing static knowledge in the Knowledge Base, the dynamic data in the Working Memory and the processing in the Inference Engine. The fourth element, the Interface, allows users to insert data, interact with the application and obtain the outputs.

Naturally the complete metaheuristic process includes other features, not addressed here, to suite SINGRAR's decision context. For instance, when tasks have priorities associated (e.g., equipment repairs), instead of the analysis being done in a unique batch which encompasses all tasks, the resource assignment problem can be done sequentially for each task, by descending order of individual priority, or in sub-batches, by descending order of class of priority.

12.3.3 Forward and Backward Chaining in the Inference Process

As referred before SINGRAR is an Expert System. Expert systems differ from decision support systems mainly in their capability of offering explanations on the advice regarding the recommended lines of action [16].

The importance of such a feature results from the effect of "bounded rationality" that affects decision-makers when they are exposed to complex problems, with large amounts of information, usually vague and incomplete, which exceeds their quite limited capacity for processing information and assessing alternatives [7, 14], leading humans to take decisions based on a 'satisficing' approach rather than an 'optimizing' approach [14]. This is the typical context for most emergency management operations where the reasoning process is complex, the number of decision factors is high and when there is a strong possibility of users having limited awareness about the context [12, 15]. In such situations the advice may become obscure and explanatory information detailing the reasoning process adopted will help decision-makers judge if advice is good or if must be discarded. Obviously the adoption of the advised actions is not mandatory, and the decision-maker has to evaluate and validate system's recommendations.

SINGRAR inference chain is traveled in one direction or the other depending if the inference engine is processing data to provide advice or if is presenting the arguments that explain how an advice was produced, as illustrated in Fig. 12.16 for the repair priority decision-making process. Forward chaining is used to generate advice, when the inference process evaluates IF THEN rules, generating conclusions based on the status of the conditions. The same rules are used in a backward chaining process (from conclusions to criteria) to explain/clarify the conditions assessed and the status of the decision elements that led to a particular advice.

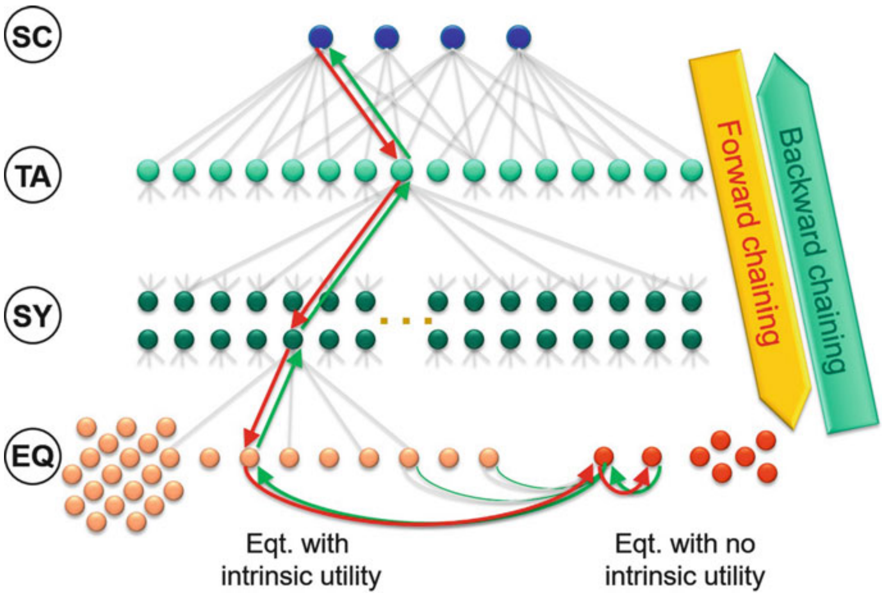


Fig. 12.16 Example of the forward chaining inference process used to compute the repair priority of equipment, and of the backward chaining inference process used to explain the set of criteria and context status that led to the advice

12.4 System Usability

Usability importance is particularly high when systems are complex, and the accuracy and timeliness of operation is decisive to the system usefulness. Thus, the usability of Emergency Management systems like SINGRAR is of utmost relevance.

SINGRAR was developed from the very beginning bearing in mind usability concerns, particularly because the system is operated using a quite high number of interfaces, some of them presenting a significant density of complex information. Designing a high usability and reliability application was very challenging considering the requirements set that users must be able to use the system effortlessly, with minimal training and to perform the tasks in the shortest time possible.

As discussed in Sect. 11.2 of Chap. 11 (System Requirements Analysis), SINGRAR development adhered to ISO recommendations for human-centered design of computer-based interactive systems. In the previous section were discussed issues related with the implementation stage of the user-centered design cycle.[2] The present section addresses the last stage of the cycle, corresponding to the evaluation of the design against the requirements, describing the assessment conducted at the end of the development of the advanced distributed prototype. The assessment study was conducted with the help of Usability experts from Academia. A detailed description of this assessment is available at [9].

ISO 9241 refers that usability is measured as a function of the degree to which the goals of the system are achieved (effectiveness), of the resources (such as time, money, or mental stress) that must be spent to achieve the objectives (efficiency) and of the extent to which users of the system find it acceptable (satisfaction) [4].

Therefore, the initial goals set to the SINGRAR assessment were to: identify factors affecting operators' effectiveness, efficiency and satisfaction; recommend potential solutions to improve SINGRAR; and assess the gains achieved by the implementation of improvements. For this purpose the team of analysts, with the support of SINGRAR development team, gathered the necessary data.

During data collection a significant sample of users within the target population was observed directly by analyst while performing activities that reproduced typical operation situations. The group of users engaged in the assessment was heterogeneous, either in terms of operation experience and domain of expertise. To verify if the peculiarities of the work environment could affect the reliability of the usability study, some sessions were performed with the operators wearing the personal protective equipment used in emergency, namely the gloves.

Data collection was conducted in 12 sessions, where users had to operate the system performing a set of tasks listed on a predefined and validated script. The script included nine activities composed of ten tasks each. The data collection procedure was designed to evaluate the efficiency, effectiveness and satisfaction of system operation and also to compare different operational methods (Table 12.1).

[2]Refer to Fig. 11.8 of Chap. 11.

Table 12.1 Characteristics assessed in the SINGRAR usability study [9]

Usability dimensions	SUMI Questionnaire (subjective data)	Activity analysis	
		Parameters	Objective data
Effectiveness	x	Number of user errors	x
Efficiency	x	Number of tasks finished in a given period of time Number of actions performed Average, maximum and minimum time for performing the activity tasks	x
Satisfaction	x	–	
Ease of memorization	x	–	

The first two characteristics were assessed using objective data collected during usability tests.

To collect and process the subjective data it was used the SUMI method [6] (in its Portuguese version [8]), which employs a metric to assess the overall satisfaction or overall usability of software, providing measures regarding the intuitiveness and ease of memorization of the application. SUMMI questionnaire was answered by the users at the end of each session.

Objective data collection included the recording of system inputs done by individual users (based on SINGRAR event log) and video recordings of the session. The video recordings were used to understand the circumstances in which the session evolved and the context of any disparate performance (e.g., long execution times and errors). The observation of the video recordings was particularly useful to isolate the methods of operation that proved problematic or, on the other hand, which constitute good practices to adopt.

The records of the Event Logs were processed to extract the main data elements (e.g., duration of each task), to detect errors in the input of the information defined by the script, and also to benchmark the progress of events.

Video recording involved two video cameras which documented the actions, comments and attitudes of users towards the application. The first recorder was placed in a fixed position perpendicular to the operator, and recorded actions, facial expressions and body posture of the users. The second camera was mobile collecting images of the computer screen, and recorded the actions performed by the users during the procedure.

The data recorded in individual user sessions were later aggregated in order to have a perception on the use of the system considering a broader set of users. In general, the data aggregation was based on average, minimum and maximum functions, which can identify trends and variability in performance. Processing the data as a whole allowed identifying the events, procedures or methods that revealed to be more problematic for the users.

12.4.1 SINGRAR Usability Analysis Using SUMI Method

SUMI method was applied to 13 users. All users answered the questionnaires correctly and their opinions were used to generate the results presented in Fig. 12.17, in terms of Median and Upper and Lower Confidence Limit, for each of the five dimensions of usability (efficiency, affect, helpfulness, controllability and learnability). These results are synthesized by the Global usability assessment.

The analysis of the results highlighted the users' positive opinion about the system (i.e. equal to or greater than the commercial standard whose reference level is 50), with some degree of dispersion in all dimensions of usability. The Global assessment with a value 60 and a small standard deviation indicates that SINGRAR is a software with high usability, better than the standard. Therefore, users were satisfied with the system, and to improve it only *ad-hoc* corrections were needed. With the exception of items related to Control that were assessed as medium (controllability = 50), all items were assessed above the reference standard. SINGRAR is perceived by users as being very useful (helpfulness = 60), satisfactory (affect = 56), efficient (efficiency = 55) and relatively easy to learn (learnability = 54). The fact that the group of users who responded to the questionnaire is sufficiently large ensures that the analysis results are relevant.

Besides the general evaluation of the system's data, it was also performed an Item Consensual Analysis (ICA). The results related with seven items of the SUMI questionnaire that departed significantly from the pattern of response expected on the basis of the SUMI standardization database. In this set of items, four reflected a positive perception by users and three reflected a negative perception. The last three items were the ones that deserved the focus of analysts and development team.

The most important issues relate with the perception on the speed of the software. Another area that deserved attention was the controllability of the system. It was

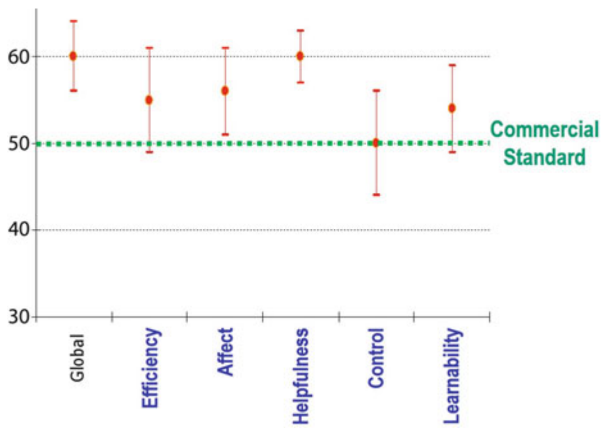


Fig. 12.17 SUMI Global Usability evaluation of SINGRAR

concluded that the results most probably reflected a lack of training on how the software works, and the recommendation was that all operators should receive training, not to feel insecure when using the system.

The use of the SUMI method offered a very good perspective about the level of quality of SINGRAR usability, and pointed to the need of implementing some minor modifications in the system, particularly in the domain of system control. In fact, this analysis together with a detailed interface and functionality analysis, allowed the identification of specific areas for improvement by the development team. After these adjustments some gains in terms of system's efficiency and effectiveness were obtained, which were evaluated and validated.

12.4.2 Dynamic Analysis of the Application

In order to analyze the aspects regarding the effectiveness and efficiency of SINGRAR, several sessions of objective data collection were conducted. As referred, the measurements taken were obtained primarily from processing the Event Log files exported from SINGRAR at the end of each session. The analysis of these logs allowed, for example, obtaining data about the time spent to perform each task, and detecting the errors in inputting the events defined in the script. The analysis was complemented by the visualization of the videos recorded, enabling the review of the circumstances in which users performed the tasks, their comments and attitudes.

The processing and analysis of data collected in each individual session allowed understanding the circumstances that led to the specific results, in terms of time spent to perform the tasks and number of errors. This analysis allowed isolating aspects of the application, of the procedures associated with data entry and of the operation methods that proved problematic, or that were good practices to adopt.

The operators engaged in the assessment sessions presented different levels of experience, in order to evaluate how this factor affected user performance. The results demonstrated that experience was not always significant in terms of the proficiency of operation. It was observed that some of the users that were supposed to be more experienced presented levels of performance worse than the inexperienced users. The causes identified for this finding related mainly with the adoption of deficient procedures for using the system.

12.4.3 Analysis of Interfaces and User Interaction

The general arrangement of the interfaces was assessed considering the specificity of the context of use. Despite some complexity of the interface, the system is designed for a very specific objective and field of application. It was found that, in general, users don't have problems accessing and using the features they need.

Another analysis of the interfaces focused on assessing the graphical user interface (GUI), considering factors such as, the standardization of symbols and methods of accessing system functionalities, the type and size of lettering, and color contrast between letters and background. Figure 12.18 illustrates the before and after of a dialog box used to insert data in SINGRAR, which is representative of the type of intervention done. The dialog box shown on the left side depicts the design that was found in the beginning of the assessment, and where some deficiencies were identified. The analysts discussed the findings with the development team and some details were modified to improve interaction. The modified dialog box is depicted in the right side of the figure. Three modifications are highlighted in the figure, which will be discussed below as examples of the intervention that resulted from the usability assessment.

Considering the image contrast there were screens lacking adequate contrast between the background and the label letters, making it difficult to read, for example, white letters over a gray background (see circles 1 and 2 in Fig. 12.18). The symbols adopted failed to adhere consistently to common standards. Sometimes the icons chosen might lead users to make mistakes or were hard to relate to the functions they were associated with (see circle 3 in Fig. 12.18). Similarly, the use of captions in buttons was not consistent, since the terminology was not always the same. It was further observed that the operators tended to assume the existence of some mouse actions common in Windows environment (e.g., double-click), which were not always programmed. Unsuccessfully trying to use such functionalities reduced efficiency and user satisfaction. The use of complete standard mouse interactions (such as double-click) was recommended and implemented.

Fig. 12.18 Examples of interface characteristics that were improved after the usability analysis. (**a**) Initial GUI. (**b**) GUI after usability analysis

For experienced users the fastest way to access the features of a program is through the keyboard, for example using shortcut keys. This option was not systematically considered in the interface design, which limited the user's efficiency, since operation often required to take the hand out of the keyboard and to use the mouse to position the cursor over a button and to click. The use of shortcut keys was recommended and implemented.

Data entry was often followed by the clicking of a command button (e.g., "OK"). Setting this button as the default button, improves user performance since it allows the operator to hit the "Enter" key in the keyboard, without having to handle the mouse. The use of default buttons was recommended for non-critical operations.

In order to expedite the selection of equipment for data entry, the "Equipment Code" (illustrated inside circle 2 in Fig. 12.18) became editable using a text box that allows writing directly the code of equipment. This solution avoided the need to manipulate other types of interfaces to select the equipment, saving significant amounts of time.

After finishing the entry of data regarding one event the software always closed the window presented in Fig. 12.18. However this form is used repeatedly to input data. A new option was offered to users, which was to save the changes resulting from the entry of data of one event and to proceed with the introduction of data from a new event, without closing the window (see new icons in circle 3 of Fig. 12.18). Besides this new feature, the cursor was also positioned automatically in the text box where the operator usually starts writing, thus avoiding the need to use the mouse to position the cursor. The implementation of these minor modifications significantly improved the efficiency in the entry of data of multiple events.

One issue that also had big influence on the systems effectiveness related with the performance of the search tool that was already available for finding equipment based on a textual descriptor. Although the algorithm worked properly, the Knowledge Base was not designed in a way that supported the system to recognize alternative acronyms (i.e., synonyms), making it difficult to find equipment that could be referred in the reports using multiple names or abbreviations. This limitation was noticed and the Knowledge Base was updated, resulting in a major reduction of the time spent by users trying to locate equipment items in the system.

12.5 Conclusions

Decision making in complex environments presents many challenges and the use of intelligent decision support tools became an invaluable asset in the toolkit of decision-makers. Naval environment is paradigmatic of such type of environments, for instance in what concerns the engineering management process when equipment repairs are needed and the ship is engaged in combat or is affected by an emergency situation. Despite experienced professionals can dispense decision support tools for many everyday tasks, such intelligent systems are a major asset and provide competitive advantage for training and simulation, and mostly for real live management of complex and high stress situations, where humans tend to fail their judgments.

The current chapter complements Chap. 11, which characterized the context of use and user and organizational requirements to comply by a knowledge-driven intelligent system designed to provide support to emergency management activities in naval environment. It presented SINGRAR which is a fuzzy distributed emergency management expert systems that provides a common platform for sharing incident status, provides support to dynamically manage priorities of alternative courses of action based on the current operational context, supports the command and control process, acts as an expert system advising lines of actions, and federates different databases which are used in conjunction with the knowledge base. The use of common robust advice tools in a distributed system ensures predictability and coherence of the decision-making process, since it supports collaborative decision-making contributing to the desired unity of effort.

Special attention was devoted to the discussion of the issues related with system design, the inference models implemented and the measures used during the project to ensure high usability standards. Regarding the inference models, the benefits of using a fuzzy logic to develop the approximate reasoning solutions were argued considering the complexity of the problems handled and the vagueness and imprecision of the data processed. The adequacy of the model is confirmed by the very robust advice provided, which justify SINGRAR operational validation and a decade of use onboard of two Portuguese Navy frigate classes.

Some of the advantages that SINGRAR offers are: instantaneous integration of the information compiled at different workstations; automatic and coherent reaction to scenario changes; fault tolerance; increased survivability of compiled information and decision support capabilities; and decrease of total time between acknowledging an incident and the triggering of the response, thus improving tempo, information sharing, situational awareness, responsiveness and coordination.

The Portuguese Navy plans to develop a second generation of SINGRAR, benefiting from the technological evolution that happened along the way. As the author argued in some of references used in this chapter, SINGRAR model can be applied to other crises response contexts, namely to support disaster relief operations in an inter-agency scenario. This is the challenge that the author embraced recently in the lead of the project THEMIS (dis**T**ributed **H**olistic **E**mergency **M**anagement **I**ntelligent **S**ystem), which is a Portuguese Ministry of Defense funded R&D project developed by a consortium of Portuguese Armed Forces Services, Academia and Private sector entities.

References

1. Carley, K.M.: Stopping adaptation. In: CASOS Conference 2002 (2002)
2. Carley, K.M., Lee, J.-S., Krackhard, D.: Destabilizing networks. Connections **24**(3), 79–92 (2002)

3. Chen, S.-J., Hwang, C.-L.: Fuzzy Multiple Attribute Decision Making: Methods and Applications. Lecture Notes in Economics and Mathematical Systems, vol. 375. Springer, Berlin/Heidelberg (1992)
4. ISO: ISO9241 Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on Usability. International Organization for Standardization (1998)
5. Joslyn, C.: Link analysis of social meta-networks. In: CASOS Conference 2002 (2002)
6. Kirakowski, J.: The use of questionnaire methods for usability assessment (1994). http://sumi.ucc.ie/sumipapp.html. Accessed Feb 2016
7. Miller, G.A.: The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychol. Rev. **63**, 81–97 (1956)
8. Nunes, I., Kirakowski, J.: Usabilidade de interfaces - versão portuguesa do software usability measurement inventory (SUMI) [interfaces usability - Portuguese version of the software usability measurement inventory (SUMI)]. In: Arezes, P., Baptista, J.S., Barroso, M.P., Carneiro, P., Cordeiro, P., Costa, N., Melo, R., Miguel, A.S., Perestrelo, G.P. (eds.) Occupational Safety and Hygiene–SHO 2010, pp. 389–393. Portuguese Society of Occupational Safety and Hygiene (SPOSHO), Guimarães (2010)
9. Nunes, I.L., Simões-Marques, M.: SINGRAR usability study. In: Design, User Experience, and Usability. Design Philosophy, Methods, and Tools: Second International Conference, DUXU 2013, Held as Part of HCI International 2013, Las Vegas, NV, July 21–26, 2013, Proceedings, Part I, pp. 359–368. Springer, Berlin/Heidelberg (2013)
10. Panzarasa, P., Carley, K.M., Krackhardt, D.: Modeling structure and cognition in organizations: a meta-network computational approach. In: CASOS Conference 2001 (2001)
11. Simões-Marques, M., Nunes, L.I.: A fuzzy multicriteria methodology to manage priorities and resource assignment in critical situations. In: Humanitarian and Relief Logistics: Research Issues, Case Studies and Future Trends, pp. 129–153. Springer, New York (2013)
12. Simões-Marques, M., Pires, F.J.: Singrar—a fuzzy distributed expert system to assist command and control activities in naval environment. Eur. J. Oper. Res. **145**(2), 343–362 (2003)
13. Simões-Marques, M., Ribeiro, R.A., Gameiro-Marques, A.: A fuzzy decision support system for equipment repair under battle conditions. Fuzzy Sets Syst. **115**(1), 141–157 (2000)
14. Simon, H.A.: Theories of bounded rationality. In: McGuire, C., Radner, R. (eds.) Decision and Organization, pp. 161–176. North-Holland Publishing Company, Amsterdam (1972)
15. Simpson, N.C., Hancock, P.G.: Fifty years of operational research and emergency response. J. Oper. Res. Soc. **60**, S126–S139 (2009)
16. Turban, E., Sharda, R., Delen, D.: Decision Support and Business Intelligence Systems, 9th edn. Prentice Hall Press, Upper Saddle River (2010)
17. Zadeh, L.: The concept of a linguistic variable and its application to approximate reasoning - III. Inform. Sci. **9**(1), 43–80 (1975)
18. Zadeh, L.: The concept of a linguistic variable and its application to approximate reasoning - I. Inform. Sci. **8**(3), 199–249 (1975)
19. Zadeh, L.: The concept of a linguistic variable and its application to approximate reasoning - II. Inform. Sci. **8**(4), 301–357 (1975)
20. Zimmermann, H.-J.: Fuzzy Set Theory—and Its Applications. Springer Netherlands, Dordrecht (2001)

Chapter 13
Crop Protection Online—Weeds: A Case Study for Agricultural Decision Support Systems

Mette Sønderskov, Per Rydahl, Ole M. Bøjer, Jens Erik Jensen, and Per Kudsk

Abstract Crop Protection Online—Weeds (CPO-Weeds) is a decision support system for weed control developed in Denmark and later adjusted to conditions in several other countries. In Denmark, the DSS includes all major crops and available herbicides. The background for developing CPO-Weeds was a political motivation for reducing pesticide use and the concept of factor-adjusted doses. It was never the intention to build a sophisticated scientific model, but rather to design a simple user-friendly system. It is a knowledge-driven DSS, which offers herbicide dose suggestions based on a large database of the existing knowledge of herbicides and herbicide efficacies. The required weed control level in CPO-Weeds is based on expert evaluations, a herbicides dose-response model and an additive dose model to calculate possible mixtures of herbicides targeted a specific weed population. The herbicide dose model is a two parameter dose-response model, which is modified to include the effects of temperature, weed growth stage and influence of drought. The development has been driven by an ambition of offering a robust system with relatively low amounts of input variables and limited need for experimental parameter generation. CPO-Weeds offers overview and guidance for field specific spraying solutions, and the system has proved able to recommend herbicide doses with considerable reductions compared to label rates. Furthermore, CPO-Weeds offers a variety of tools that summarises knowledge of herbicides for a wide range of questions asked by practical weed managers, e.g. efficacy profiles of each herbicide, efficacy of users own herbicide mixtures, weed identification key and guidance for spraying strategy. The experiences have shown that even though CPO-Weeds are considered robust and trustworthy by both farmers and advisors there is a relatively low number of farmers subscribing to the system. A survey revealed that the DSS

M. Sønderskov (✉) • P. Kudsk
Department of Agroecology, Aarhus University, Forsøgsvej 1, DK-4200 Slagelse, Denmark
e-mail: mette.sonderskov@agro.au.dk

P. Rydahl • O.M. Bøjer
IPM Consult Ltd, Hovedgaden 32, DK-4295 Stenlille, Denmark

J.E. Jensen
SEGES P/S, Agro Food Park 15, DK-8200 Aarhus N, Denmark

© Springer International Publishing Switzerland 2016

J. Papathanasiou et al. (eds.), *Real-World Decision Support Systems*, Integrated Series in Information Systems 37,
DOI 10.1007/978-3-319-43916-7_13

falls in between the strategies of many farmers; either the farmers relies completely on own experiences or advisory services or they considers the full crop rotation in their weed management. The latter is not supported by CPO-Weeds, which focus on a single season. The long term consequences of herbicide recommendations is only included in the need to limit input to soil seed bank. Another limiting factor for an increased practical use of CPO-Weeds is the need for field monitoring of weed populations, which can be a time consuming task and requires extensive weed recognising abilities of the farmer at the very early growth stages of weeds. The intention of CPO-Weeds was to provide recommendations for the full spraying season of a field, but experiences have shown that the system has several uses. Many farmers spray with a standard solution in the autumn in winter crops and then use the DSS for spring sprayings. The relatively simple input requirements also make the DSS suitable for teaching purposes and for farmers starting to grow new crops in their rotation as a learning tool.

13.1 Introduction

Weeds are detrimental for crop production if not managed properly. Globally, the potential yield loss caused by weeds can be as much as 34 %, and weeds are the pest group causing the largest yield losses [13]. Weeds are a diverse group of plant species and a broad variety of herbicides are available for their control. Even though many of the oldest and environmentally most toxic herbicides have been banned, there are good reasons for minimising the applied amounts to minimise the adverse impact on human health and the environment. Conventional farming rely strongly on pesticides to protect the crops from pests, but recognizing the necessity for a reduced reliance on pesticides is forcing farmers to adopt the concept of integrated pest management (IPM). IPM implies that preventive strategies and non-chemical control methods are integrated into the pest management strategy [1]. Pesticides should only be applied when necessary and then only in the dose required to control the pest. This complicates management decisions, but increases sustainability of the agricultural practices. The conversion to IPM is proceeding slowly in European agriculture. One reason is that an incentive is needed to diverge from current practice. Pesticides are often the cheapest weed management control measure, and they are easy to use with high efficacy and relatively little variation. The widespread evolution of pesticide resistance is, however, a powerful driver for farmers to start practicing IPM. A first, fairly uncomplicated step, towards IPM is applying pesticides according to the principle: as little as possible—as much as required.

Crop Protection Online—Weeds (CPO-Weeds) is a knowledge-driven decision support system (DSS) for chemical weed control and offers support for herbicide and dose selection in a variety of crops [9, 14]. CPO-Weeds suggests management options based on a large database comprising existing knowledge on herbicides and herbicide efficacies. There are pre-emergence and post-emergence herbicides.

The pre-emergence herbicides, generally, control a broad spectrum of weeds. These herbicides are applied before the actual weed composition in a field can be determined and the application relies on knowledge of previous year's weed infestations and general recommendations. The post-emergence herbicides, on the other hand, can often be targeted much more specifically and there are significant potentials for reductions in herbicide use in post-emergence herbicide applications. Herbicide reductions are, however, only possible with due consideration of the actual weed flora of individual fields combined with information on the competitiveness of the crop and climatic factors like temperature or drought. This information is required due to the highly variable susceptibility of weed species to the various herbicides. Standard herbicide recommendations based only on crop, timing and geographic location tend to suggest broad spectrum solutions at high doses in contrast to tailor-made recommendations adapted to individual fields. The standard solutions are intended to control the majority of weeds, irrespectively of growth stage and density under various climatic conditions.

Previously, before the availability of any DSS for weed control, farmers had no choice but to rely on the advice from advisors or simply on the label information on the at-hand herbicides. The situation is presently more diverse in Denmark, but the majority of advisory services publish general advice for specific crops or, more commonly, regional advice based on the most common weed species. The advisors have limited time for providing specific recommendations for individual fields, or the cost for such advice is too high for many farmers. This often results in application of herbicide mixtures that can control a wider range of weed species than present in many fields and in higher doses than necessary considering the actual weed population. Both farmers and advisors can benefit from the use of a DSS for weed control. The farmer can get specific solutions for each field if the weed flora is monitored, and the advisor can make a set of more variable advices with limited time consumption. The latter can aid a diversification of standard recommendations. Furthermore, a DSS is an excellent learning tool for many advisors, as it shows e.g. weed spectra covered by individual herbicides and herbicide mixtures.

Several DSSs for weed control have been developed throughout the world. In 2009, a report summarised the availability of DSSs in Europe [2]. The conclusion was that nine systems were available in Europe for weed management with large differences observed among the systems. The aims of the DSSs ranged from providing simple herbicide solutions using labelled rates, to DSSs providing a list of solutions with dose reduction potential and options for optimizing for either environmental impact or economic output. Two systems focused solely on one factor, either the substitution of one herbicide with another according to their environmental impact or the optimal timing of herbicide application, distinguishing them from the other DSSs. The number of crops implemented in the nine DSSs ranged from 1 to 30, with the majority in the range of 1–5. Four systems contained from 4 to 30 crops. Of these four systems, only CPO-Weeds and the Dutch DSS, Minimum Lethal Herbicide Dose (MLHD) [3, 6] have been validated in short term field experiments. Both systems rely on monitoring the actual weed population in the field, including the size and frequency of the species. MLHD further requires a specific measuring device as the need for follow-up herbicide applications is determined based on sensor measurements. MLHD works only for herbicides affecting the photosynthetic activity.

CPO-Weeds offers a validated approach for 30 crops, 110 weed species, all registered herbicides in Denmark, and the DSS is available online with regular updating. CPO-Weeds is knowledge-driven and based on a herbicide dose model driving the herbicide and dose selection, whereas the thresholds and required weed control levels are expert evaluations that have been adjusted according to validation trials and user's experiences over more than 25 years. The following chapter sections describe the background for the development of CPO-Weeds, system structure, available tools, user interface, implementation, the experiences with the system and the main conclusions.

13.2 System Requirements Analysis

An increased political focus on reducing pesticide application was the driving force for developing the first version of CPO-Weeds in the 1990s that could aid the farmer's spraying decisions with reduced herbicide doses. The Danish Parliament passed the first Pesticide Action Plan in 1986 and in the following decades new action plans have continuously been implemented with the objective to reduce pesticide use and promote more specific and targeted pesticide recommendations. An indicator, the Treatment Frequency Index (TFI), was introduced to monitor the development in pesticide use from year to year. Every herbicide was assigned a standard dose for each recommended use. TFI for a specific pesticide use is calculated as the ratio between the applied dose and the standard dose [4]. In order to meet the goals of the pesticide action plans, the amount of pesticides applied had to be reduced. The concept of factor-adjusted herbicide doses, implying that the herbicide dose is adjusted according to factors affecting herbicide efficacy, e.g. weed species, growth stage and environmental conditions, was adopted by many farmers [7].

Before the introduction of CPO-Weeds, this adjustment of herbicide doses relied on the experiences of farmers and advisors. To exploit the full potential of reduced herbicide doses, and to combine large numbers of factors (crops, weeds, herbicides, climatic conditions) a DSS for evaluating the optimal dose and herbicide for specific situations was needed. Hence, the primary functional requirement, which was identified during the initial developmental phase, was to provide field specific recommendations for herbicide doses considering the observed weed population and the most important influencing factors. Another functional requirement was to provide farmers and advisors with tools to enhance the overview of options for herbicide application. A strictly scientific approach was not adopted developing CPO-Weeds. It was never the intention to build a research tool, rather the objective has been to provide a practical management tool with as few input requirements as possible, and algorithms that were simple enough to be parameterised for all major crops and weed species. More detailed mechanistic models have high requirements for parameters, and it is difficult and expensive to gather the required information to keep such systems updated. CPO-Weeds build on relatively simple algorithms, which scientifically might be improved by replacing the underlying equations with more accurate ones. The system requirement analysis also identified non-functional requirements, such as robustness of recommendations, minimisation of input requirements, compliance with legal restrictions, inclusion of all available herbicides, maintainability and accessibility to interact with developers for the users.

Throughout the development of CPO-Weeds the ability to parameterise the model has been a key point. The number of herbicides available is relatively high in the major crops; hence the amount of work to produce the necessary parameters can be high. To maintain parameter input at an acceptable level only few equations are included in CPO-Weeds. Therefore, the practical validation and input from users have had high importance to validate the model performance and confirm that this simple approach is valid. The main focus of CPO-Weeds has not changed during the development although the functionality of the system has developed.

13.3 System Design and Problem Solving Technic

CPO-Weeds encompass a variety of tools, but the core tool is the "Problem Solver", which meets the requirement for specific herbicide recommendations for control of a specific weed population in a field. The structure of the DSS provides opportunity for the data, generated for the "Problem Solver", to be stored in a common database and used in the tools, such as the "Efficacy Profiles", "User's Mixture", "Herbicides Across" and "Efficacy tables". As these tools are powered by a joint mathematical model [the dose-response model with factor adjustments, described in Fig. 13.1 and Eq. (13.1)], the integrity of output from these tools is automatically ensured. The tools sort and present the available herbicide solutions in different way as a response to the questions asked by the user. Several names for the system have been used in the English literature, including PC Plant Protection (Denmark), Plant Protection Online (Denmark), VIPS-Ugras (Norway), CPOWeeds (Spain) and DSSHerbicide (Poland, Germany) [11, 12, 18]. In each country, the CPO-versions are adjusted according to the herbicide availability in the specific country and parameterised accordingly, whereas the algorithms and calculations follow the same concept. Not all tools are/have been available in all countries and the number of crops and weed species differ among the DSSs. VIPS-Ugras in Norway and DSSHerbicide in Poland and Germany are currently available online, whereas the Spanish CPOWeeds is used as a practical research tool.

Problem Solver
Herbicide recommendations in CPO-Weeds are calculated through a three step process following the user's input on weed composition in the field [9, 14]. The first step is to determine whether the weed infestation requires control. The threshold for each weed species depends on crop, crop growth stage and crop density plus

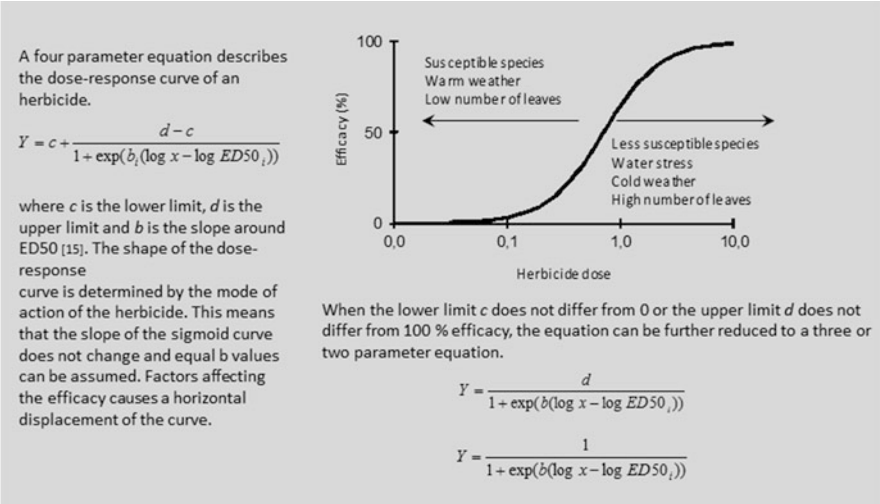


Fig. 13.1 A general description of the dose-response model and the factors affecting herbicide efficacy

the growth stage and density of the weed species. The thresholds were initially based purely on expert knowledge and evaluation, but the thresholds have been adjusted according to practical experiences during the years. The second step of the decision process is the selection of herbicide solutions that can provide the required control of each of the weed species. The third step is the calculation of potential herbicide mixtures that can control the whole weed spectrum. In addition to the three steps there is another step, which determines the need for adjuvant based on general recommendations. The three first steps are explained in more detail below. For an overview of the steps, which the DSS runs through to calculate a list of recommendations, see Fig. 13.2, which shows the steps in a schematic form from the start of processing (get parameters from user input) to the display of possible solutions and end of processing. Further reference is made to Fig. 13.2 in the following.

Determining Control Needs

All weed species have been assigned a required control level for each crop, which is decisive for the herbicides dose recommended by CPO-Weeds ("Get required control levels", Fig. 13.2). The required control level was determined by experts based on the available knowledge of weed and crop competitiveness, expected yield losses and weed seed production. Growth stage of the crop is described by the BBCH scale [10]. The user can choose between three levels of expected yield; an intermediate yield typical for Denmark and a higher and a lower level. Cropping conditions are thus divided into categories that cover the range observed under Danish conditions providing enough information for herbicide solutions to be defined. Higher weed control is needed when the expected yield is low caused by the less dense canopy.

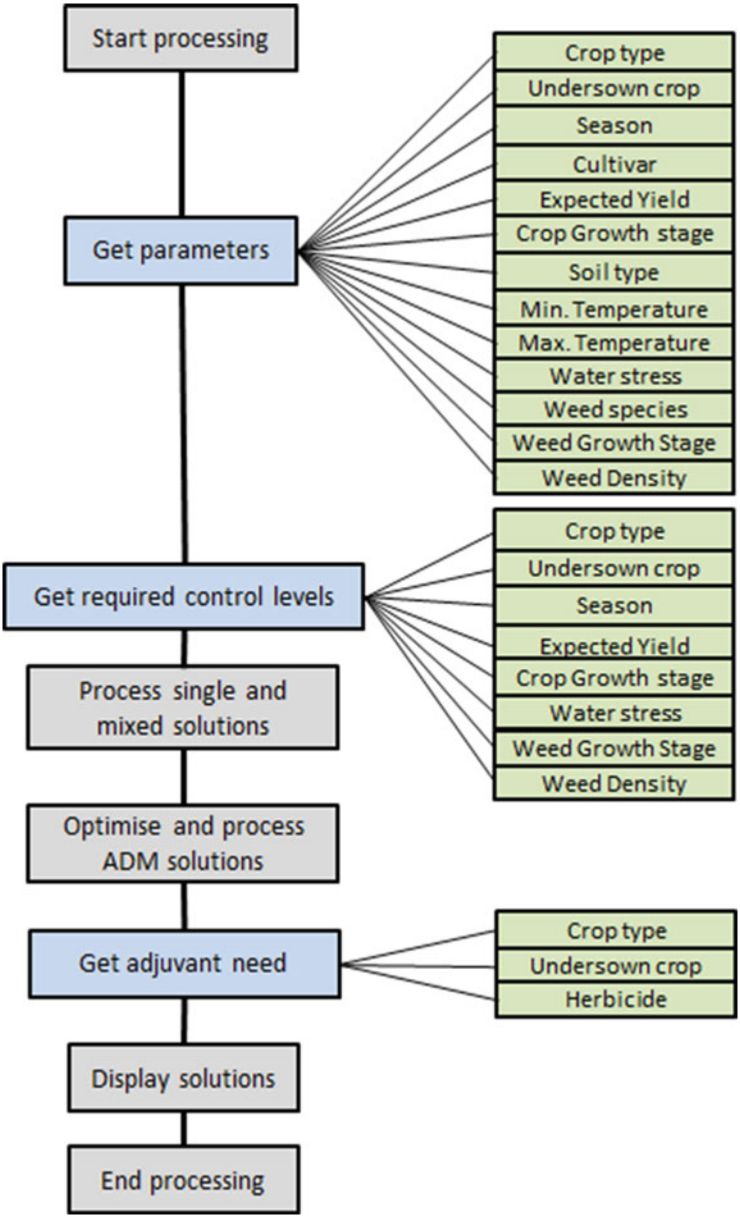


Fig. 13.2 Schematic overview of the processes in the tool "Problem solver". *Grey boxes* are webserver processes, *blue boxes* are database lookup (stored procedures) and *green boxes* are parameters from the user interface. The three steps in the tool "Problem solver" is Step 1:"Get required control levels" where the level of control required for each weed species in the field is acquired from the database table. Step 2 + 3: "Process single and mixed solutions" and "Optimise and process ADM solutions", where the DSS provides all potential solutions that can control the actual weed flora and optimise them according to costs, Treatment Frequency Index (TFI) or Pesticide Load (PL)

Inversely, at high yield levels competition will be higher and lower control levels can be accepted. Equally important for determining the requirement for control is the composition of the weed flora and the growth stage of the weeds (Fig. 13.2). A high weed density will increase the required control level. The individual weed species have different competitive abilities; therefore they require different control levels. In a dense crop with high yield expectance and low weed density, or weeds with low competitive ability, the DSS may well end up recommending no herbicide application. For winter annual crops the required control level in autumn and spring varies under Northern European climatic conditions. Economic thresholds only consider yield losses, while additional parameters are important considering the long term management of weeds, e.g. reducing/avoiding a build-up of the weed population. Such considerations are taken into account in the expert assessments of required weed control levels in CPO-Weeds. No actual model lies behind these thresholds; they are simply the outcome of consensus among experts and years of practical validation. If no spraying is recommended, CPO-Weeds will advise the user to repeat the weed monitoring at a later time to handle a potential new flush of weeds.

Identifying Possible Solutions for Individual Weed Species

Having determined the requirements for control, the DSS proceeds to the second step and generate a list of herbicides solutions that can control the weed flora present in the field (single solutions, Fig. 13.1). The dose of all relevant herbicides that provide the required level of control of each of the present weed species is calculated. The calculations also take into account the climatic conditions on the day of spraying and any soil moisture deficit [Fig. 13.1 and Eq. (13.1)].

The general concept of dose-response curves is explained in Fig. 13.1 and by incorporating the adjusting factors into the two-parameter model the herbicide model used in CPO-Weeds is evolved [Eq. (13.1)].

$$E_n = \frac{100}{1 + exp\left(-2\left(a_n + b_n\left(log\left(\frac{d}{r_s \times r_t \times r_v}\right)\right)\right)\right)} \tag{13.1}$$

where E_n is the relative efficiency on weed n in percentage, d is the dose, a_n is the horizontal displacement of the curve for weed n, b_h is the slope of the curve at ED_{50} (the dose that provides 50 % efficacy for weed n), r_s quantifies the differences among weed growth stages, r_t quantifies the differences among temperatures and r_v quantifies the differences among levels of drought.

The basis of this calculation is the parameterisation of individual dose-response curves for all relevant herbicide-weed species combinations. The parameterisation assumes parallel dose response curves for herbicides with the same mode of action, an assumption that has been shown to be reasonable for the use in the DSS [7]. This means that the ratio of doses inducing the same efficacy level for different weed species is constant [8] (Fig. 13.1). More accurate approaches are available, but applying those increases the requirements for data for parameterisation. The variability of empirical data is high and this variability must be accounted for in the parameterisation. Thus, the dose-response model calculated from the available background data is evaluated and if the variability in the data is large, a safety margin is established. This implies moving the dose-response curve slightly to the left thereby decreasing the efficacy of a given herbicide dose. Albeit not a scientific approach, it has ensured the robustness of CPO-Weeds. The least susceptible weed species in a specific field will determine the herbicide dose needed to achieve the required control level of all weed species present in a field. This implies that more susceptible weed species will be controlled more efficiently than required. This is adding to the robustness of CPO-Weeds.

The parameterisation of the dose-response curves has been possible due to the access of numerous data from herbicide efficacy experiments with more than one dose. The majority of the data originates from field efficacy trials conducted by Aarhus University on behalf of the agrochemical companies. Even though a substantial amount of data was available for the parameterisation of dose-response curves, there are data gaps. In some cases, the gaps can be filled by expert estimates or by grouping weed species with common characteristics. Typically, there is a lack of data in minor crops, where there is no economic incentive to authorize herbicides. Data gaps cannot be fully avoided. CPO-weeds may therefore be perceived as a system that integrate and interpret current knowledge according to conditions on a field level and present this to the user.

Estimating Optimum Herbicide Tank-Mixtures

The third step is driven by the Additive Dose Model (ADM) [7, 17]. ADM combines available herbicides into herbicide tank-mixtures consisting of up to four different herbicides at specified doses (mixed solutions from Fig. 13.2). In fields where only one weed species requires control, one herbicide can often solve the problem, but when more weed species are present this may not be the case. The ADM estimates the optimal composition of herbicide mixture, and the doses of each of the herbicides minimizing either the costs, the TFI or the pesticide load (Fig. 13.3). The optimisation is performed using linear optimisation or linear programming (the name of the mathematical method). ADM assumes that the efficacy of the herbicides is additive, hence no synergistic or antagonistic effects are included. If herbicide mixtures are known to act antagonistic, they will not be recommended by CPO-Weeds. If herbicide mixtures are synergistic, as very rarely is the case, they will perform better than predicted by CPO-Weeds. The list of recommended herbicide mixtures are controlled via a database that precludes any non-legal mixtures and mixtures not recommended by the agrochemical companies or advisory services. The last action before displaying the recommendations to the user is inclusion of adjuvant to the herbicide solutions if needed ("Get adjuvant need" Fig. 13.2).

Other Tools

The variety of tools available in CPO-Weeds is based upon the same parameterisation of dose-response curves, and simply summarises the data in various useful ways. Basically, the program uses the same database parameters in all of the tools. "Efficacy profiles" provides the user with an overview of the susceptibility of the weed species to each of the herbicides included in CPO-Weeds. "User's

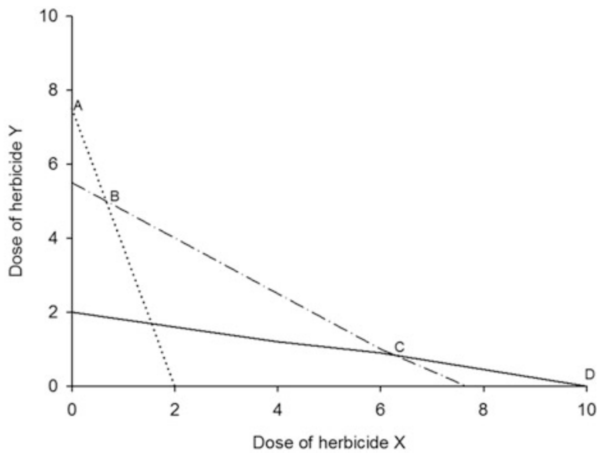


Fig. 13.3 Illustration of the ADM for two herbicides and three weed species. The *lines* represent the possible combinations of the two herbicides providing a predetermined level of control. The combinations marked A–D will all represent herbicide mixtures that can provide the required level of control or more of the three weed species. The optimisation based either on cost, TFI or pesticide load will determine which solution is listed first. CPO-Weeds offers solutions with up to four different herbicides in tank-mixtures. *Solid line*: weed species 1, *dashed line*: weed species 2, *dotted line*: weed species 3

mixture" enables the user to find the efficacy of a mixture defined by the user. "Herbicides Across" provides overview of available herbicides for a specific crop or weed species and lists the uses of the herbicides. "Efficacy tables" summarises the efficacies of all herbicides against all weed species for a specific crop. "Strategy for a growing season" does not use the database parameters, but offers guidance for the timing of herbicide applications related to weed and crop development through the season. The broad variety of tools is offered to facilitate as many applications of CPO-Weeds as possible. Thus CPO-Weeds can be perceived both as a tool for solving specific weed problems and a learning tool providing the user with information on the parameters influencing herbicide activity and their interaction. CPO-Weeds are available on https://plantevaernonline.dlbr.dk by subscription.

System and Language Specifications

CPO-Weeds runs on two different servers; a web server and a database server (Microsoft SQL Server) The webpages are hosted in the IIS framework using Active Server Pages technology (ASP) for some webpages and ASP.NET for other web pages. The server scripting language for ASP pages is Jscript and the server programming language for ASP.NET pages is C#. On the client, the scripting language is JavaScript in both cases. Server side scripts/programs are connected to the database(s) using ADO or the SqlClient library of the .NET framework.

13.4 User Interface Design

The input variables, which are included in the present version of CPO-Weeds, are crop type, crop density, crop growth stage, weed species and weed growth stage along with season, temperature on the day of spraying, any soil moisture stress in the field and the presence of known resistant weed biotypes. In order to simplify the input of data the start page for each tool has been kept to maximum of one screen page (Fig. 13.4 shows the input page for "Problem solver"). The output page might be longer due to the list of possible recommendations (sections of output pages are shown in Figs. 13.5 and 13.6). The relatively low number of input variables is essential for any DSS developed for farmers. The system needs to be easily assessable and only require a minimum of input as the users are not necessarily interested in spending long hours at the computer. The ability to have all inputs visually present before asking for the final recommendation makes it easier to understand the connection between inputs and outputs. The user can easily go back and change an input variable to understand the impact on the recommendations. It is, for example, possible to change the density or growth stage of a certain weed species and immediately see the impact on the recommended doses. This increases the

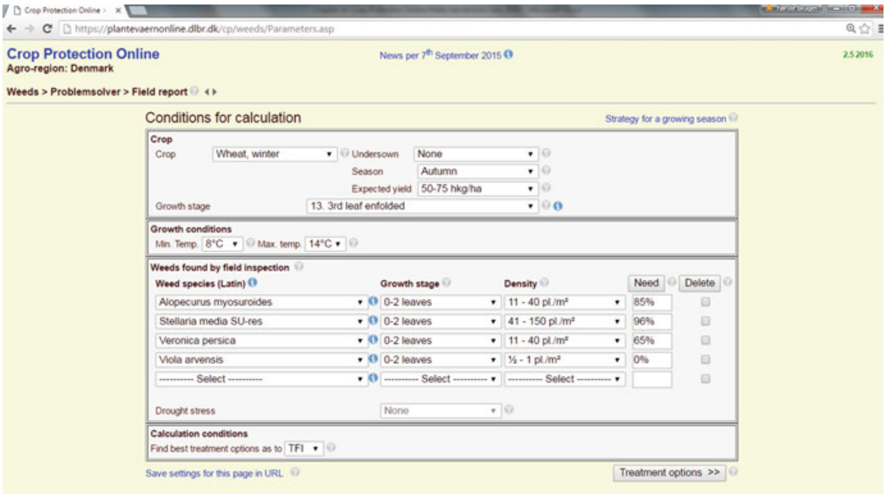


Fig. 13.4 The user interface for the tool "Problem Solver" with an input example from a winter wheat field. It is an autumn spraying with the crop in growth stage BBCH 13, expected minimum/maximum temperatures on the day of spraying of 8/14 °C, no previous drought and four weed species in different densities all in growth stage 0–2 leaves. The required level of control in this situation is indicated for each species and cannot be changed by the user. In this case, recommendations are optimized according to TFI. The density of the weed species *Viola arvensis* is lower than the threshold for this species, hence no need for control. The other weed species have control requirements between 65 and 96 %. As two problematic weed species are selected; *Alopecurus myosuroides* and a resistant biotype of *Stellaria media*, the required control levels are relatively high

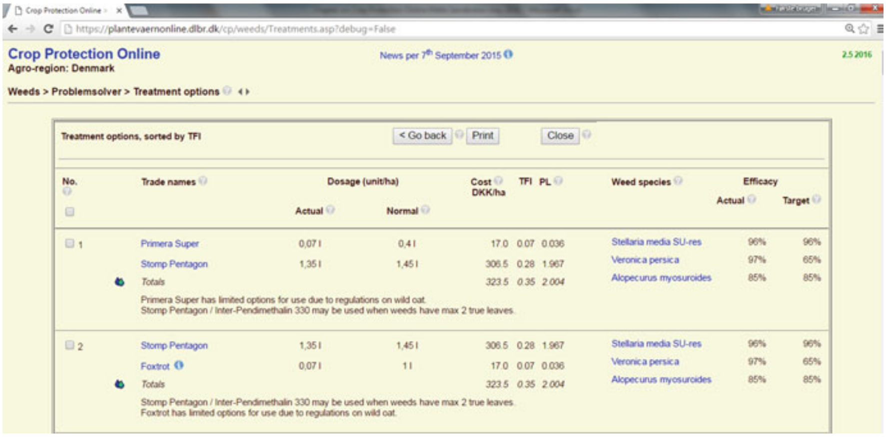


Fig. 13.5 The output page with recommendations for the field situation in Fig. 13.4. The topmost two solutions are shown. The solutions were optimised based on TFI. The recommended dose is termed the "Actual dose" and the "Normal dose" is the label rate. The treatment frequency index (TFI), the cost per hectare, the pesticide load (PL) and the estimated efficacy levels for the weed species are shown. The "Actual efficacy" is the efficacy estimated from the recommended dose, whereas the "Target efficacy" is the required control level for the individual species. This means that some weed species will be controlled more efficiently than needed, but the species with the highest requirement for control will decide the recommended dose. In the first solution the control of *Veronica persica* is estimated to 97 % because this species is susceptible to the same herbicide (Stomp Pentagon) as the other broadleaved weed species, the resistant biotype of *S. media*. Primera Super is controlling the grassweed *A. myosuroides*

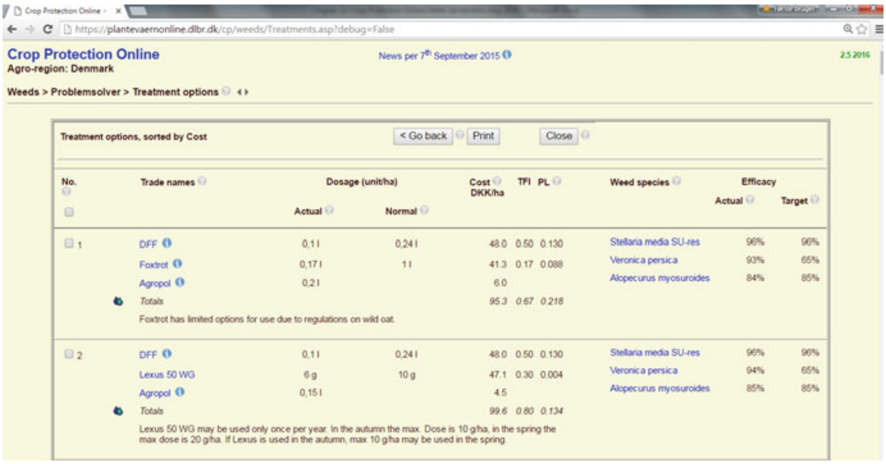


Fig. 13.6 The output page with recommendations for the field situation shown in Fig. 13.4, but now optimised according to cost per hectare. The optimal solution when optimised according to TFI had a cost of 323.50 DKK per hectare. Adequate efficacy can be achieved for 95.3 DKK per hectare if TFI is neglected. In this example, it will have a high economical consequence for the farmer to consider the lowest TFI, but there are several other intermediate solutions if the user decides to consider both cost and TFI

learning potential of the DSS and visualise the importance of herbicide application timing. User interfaces for the other tools are not shown.

13.5 System Implementation

CPO-Weeds is based on a relatively simple herbicide response model, which can generate a very large amount of dose-response curves for numerous combinations of weed species and herbicides with a reasonable requirement for parameter inputs. An advantage of CPO-Weeds is that it ensures that the user will only be advised legal and effective recommendations from the huge database of dose-response curves. Every herbicide has authorized uses described on the label. This includes time of the year, crop growth stage and maximum herbicide doses. This information is built into CPO-Weeds and works as a filter sorting the recommendations. For example, choosing "autumn" as the season will cut of the herbicides that are only authorized for use in the spring and vice versa. Herbicides are only authorized for specific crops at certain growth stages for numerous reasons, e.g. crop safety, environmental toxicity and herbicide residues in the harvested crop. Choosing a crop at a specific growth stage will again sort out the safe and legal herbicide options. The maximum authorized dose cannot be exceeded, i.e. if higher dose is required the herbicide will not be listed as a stand-alone solution. It is, however, possible to use the herbicide in mixture with other herbicides. In addition, CPO-Weeds contains a list of useful and legal herbicide mixtures. This means that in addition to presenting the user with recommendations for reduced herbicide doses, the filtering simplifies the choices and ensures that the users are provided with safe, efficient and legal recommendations for herbicide spraying.

13.6 System User Experience

The basic algorithms have not changed over the years, and CPO-Weeds is still based on dose-response curves for herbicide efficacy. The first CPO-Weeds version strived for high efficacy (e.g. an average effect of 90 % control of all weed species in cereal crops). Field validation experiments in cereals of this early version showed very satisfactory control and in fact the control levels were so high that it indicated a potential for further dose reductions. A new version was released with more differentiated requirements for weed control. The most competitive and problematic weed species were still controlled with high efficacies, but less competitive weed species at low densities were either tolerated or partly controlled. This led to average required control levels of around 70 % including the scenarios that were below the economic threshold and did not call for any herbicide application. The validations of this version showed that CPO-Weeds maintained its robustness, and larger herbicide reductions were achieved [14, 18, 19].

Some additions/changes to CPO-Weeds have improved the DSS, while others have been found merely to increase complexity for the users without adding any real benefits. Examples are the implementation of soil type and crop variety. Both factors were expected to significantly influence the herbicide efficacy and therefore examined. If a soil is rich in organic material the efficacy of residual herbicides can be adversely affected. In Denmark, however, the majority of the agricultural land is low in organic carbon limiting the benefit of including soil type as a factor in the DSS. Crop variety might be expected to influence herbicide efficacy if the differences in weed suppression ability between the varieties is significant. The presently marketed crop varieties, however, are fairly similar in their weed suppression traits and the data required to parameterise this factor is often missing. Considering crop variety in CPO-Weeds did not introduce significant differences in herbicide recommendations. These parameters have therefore been deactivated in the current online version of CPO-Weeds, but as they are not deleted from the system they can be re-activated. This might be relevant in other countries, where these factors are more important.

When new issues arise in agriculture they should be addressed in a DSS to maintain end-user trust in the system. Herbicide resistance can reduce the level of control substantially. Even though it presently is not a widespread problem in Denmark the experience in other countries calls for immediate action. Therefore, a recent addition to the DSS is herbicide resistant weed biotypes. If the user is aware that resistant biotypes of the weed species are present in his fields, these weed species can be selected as a separate weed entity. The recommendations will account for this by only listing herbicides that are still efficient against the resistant biotypes.

Very recently, a new pesticide indicator was implemented. In Denmark, the treatment frequency index (TFI) is being replaced by the Pesticide Load (PL) and PL is now available in CPO-Weeds as a new sorting factor in line with price and TFI. In contrast to TFI, which merely reflects the intensity of pesticide use, PL is an estimate of the potential impact on human health, environmental fate and environmental toxicity. TFI is retained in CPO-Weeds as this is the index that farmers are most familiar with.

It is important that the user understand the need for all the requested input variables. The identification of key variables has played a crucial role in the development of CPO-Weeds. This is closely connected to the user friendliness and respect the fact that most users can only allocate limited time for using CPO-Weeds. The intention of CPO-Weeds has been to provide a valuable tool for farmers. The concept relies on the users to provide a detailed report of the actual weed composition in each specific field. This allows the DSS to provide efficient and solid recommendations.

The end-users have been introduced to CPO-Weeds via advisory services and their user platforms. The system has an intuitive user interface and all required actions are explained. The first versions of CPO-Weeds were distributed on CD-ROMs and the introduction of a new version was cumbersome. In 2001 the system went online and continuous updating was possible.

The original aim was to help farmers to reduce herbicide use. The experiences hitherto have shown that CPO-Weeds has potential for substantial reductions in herbicide use, if used as intended. Practical experiments in various countries have estimated reduction potentials between 20 and 40 % compared to labelled rates or standard recommendations [11, 14, 18].

The system has continuously been renewed with due regard to the user and new research findings. At all times, the main objective has been to provide the user with relevant knowledge summarized in an accessible way. The user friendliness has been in focus both in the design and how input is fed into the system. Through the many practical validation trials, the system has established itself as robust and trustworthy. The users have had open access to communication with the developers, and they have interacted to amend any shortcomings of the DSS. There has also been an ongoing dialog between the developers and the providers of data, i.e. the agrochemical companies and researchers. As not all requests from users and data suppliers can be met, the developers need to set a certain criteria for changes to be implemented. All communication from the stakeholders should be noted, and if the request is put forward repeatedly, the issue is probably worth looking into. It should be very clear how this procedure works to maintain a good cooperation with all stakeholders.

Experiences with CPO-Weeds

Despite the focus on user friendliness and the interaction with end-users, relatively few farmers use the system. Currently, CPO-Weeds has around 900 subscribers in Denmark including advisors, researchers and agricultural colleges, whereof the advisors constitute around one third. There are around 18,000 arable farmers in Denmark and around 2800 farmers with more than 100 ha [16]. In 2007, a survey among farmers was carried out to unravel some of the reasons for the relatively low number of subscribers among farmers [5]. The main conclusions were that the system did not exactly match the thinking or strategies of the majority of farmers. Based on the responses from the survey, it was possible to group farmers into three types; system-oriented decision-makers, experienced-based decision-makers and advisory contracting decision-makers. Basically, the first group is highly focussed on the long-term consequences and plan their crop rotation in detail. The second group base their management primarily on their own experiences and work by the “trial and error” concept. The third group relies heavily on interaction with advisors and are not likely to make any major decisions without the advice from outside. This leaves CPO-Weeds in a void as the DSS is based on the present season only, i.e. does not fulfil the needs of the system-oriented decision-makers. The experienced-based farmers want to test the spraying solutions with reduced doses themselves, but might use CPO-Weeds as a learning tool and a reference point. The advisory-contracting farmers are the ones most likely to rely on CPO-Weeds, but most likely through an advisor, who is then de facto the user of the DSS. Thus, even though all three groups of farmers stated a high confidence in the recommendations provided by CPO-Weeds, the DSS does not fit exactly into their way of decision-making. Other constraints for a more wide-spread use of CPO-Weeds were identified as the need for weed monitoring and the knowledge required to identify the most important weed species. Increasing farm size limits the possibility for field specific weed monitoring. Another outcome from the survey was a general lack of economic incentives for reducing herbicide doses. The latter might change in the future as pesticide taxes have recently increased substantially for the least benign groups of herbicides (based on PL). To accommodate the difficulties related to weed identification, CPO-Weeds offers guidance for weed monitoring procedure and graphical support for weed identification. The DSS relies on the user to identify the weed species at very early growth stages, preferentially at the cotyledon or first true leaves stages. This is a challenge to many farmers and a less than accurate identification can hamper the performance of the DSS.

The intention of CPO-Weeds is to provide the end-user with recommendations for tailor-made herbicide solutions throughout the season for all herbicide applications in a crop. The practical experience with CPO-Weeds has, however, shown that there are other ways of using of the DSS. One commonly referred practice in winter cereals is to follow standard recommendations for the first spraying in the autumn and then rely on CPO-Weeds for follow-up sprayings in the spring. Other reported uses include the generation of recommendations based on historical knowledge of weed populations in fields, general recommendations for a geographical region based on most frequent weed species, and recommendations based on the products kept in stock by a farmer. This shows that even though the intended use by the developers was one thing, the actual use can be another. The variability of use is supported by the range of tools offered. An example of an alternative use is a farmer, who plans a more variable crop rotation and therefore integrates one or more new crops. The limited experience of the farmer growing these new crops can be overcome by CPO-Weeds that can provide specific herbicide solutions not known to the farmer without having to involve an advisor.

For the users to gain, and maintain, trust in a DSS a continuous updating and adjustment to the real management situation is essential. Therefore, a plan for the maintenance of a DSS should be lined up from the beginning. The maintenance of CPO-Weeds relies on the fee the users pay to use the various tools of the DSS. The fee supports the continuously ongoing updating, including new herbicides and removing the ones no longer available as well as incorporating any changes in the authorisation. Finally, thresholds and required control levels may also be changed based on feed-back from the users.

13.7 Conclusions

CPO-Weeds has been available for more than 25 years, and the system is considered robust and reliable. The driver for the development of the DSS was a political agenda demanding reductions in pesticide use. This has increased the complexity of decision-making and thus induced a need for a DSS. CPO-Weeds is a relatively simple system with a high degree of user friendliness. A substantial reduction of herbicide doses can be achieved when following the recommendations of

CPO-Weeds compared to labelled rates and standard recommendations. The input required by the user is low and only variables which actually have an impact on herbicide efficacy are integrated in CPO-Weeds. This has secured the users confidence in CPO-Weeds and many years of validation and practical experience have shown high robustness and potential for herbicide reductions. The experiences, however, have shown that constructing a DSS for weed control that fits into the majority of farmer's management practices is difficult, and fewer farmers than expected have subscribed to the system. One explanation is that herbicides are still relatively cheap and very efficient. Thus the major driver for reducing herbicide doses is not to reduce costs but rather to reduce any adverse environmental impact. Increasing herbicide costs due to pesticide taxes and the increasing problems with herbicide resistance might motivate more farmers to use field specific herbicide recommendations and thereby increase the interest for a DSS like CPO-Weeds.

The use of a system like CPO-Weeds can be considered the very first step towards integrated weed management (IWM) through a reduction of the applied herbicide doses. Future perspectives for DSSs for weed control could be to include mechanical weeding solutions along with chemical solutions and to integrate herbicide resistance prevention measures. Mechanical weeding primarily works within the short timeframe of one cropping season, while herbicide resistance prevention would need to consider both previous and future weed management actions. Preventing herbicide resistance will inevitably include crop rotation and any future development of a weed management DSS must therefore consider crop rotation.

References

1. Barzman, M., Bàrberi, P., Birch, A.N.E., Boonekamp, P., Dachbrodt-Saaydeh, S., Graf, B., Hommel, B., Jensen, J.E., Kiss, J., Kudsk, P., et al.: Eight principles of integrated pest management. Agron. Sustain. Dev. **35**(4), 1199–1215 (2015)
2. Been, T.H., Berti, A., Evans, N., Gouache, D., Gutsche, V., Jensen, J., Kapsa, J., Levay, N., Munier-Jolain, N., Nibouche, S., et al.: Review of new technologies critical to effective implementation of Decision Support Systems (DSS's) and Farm Management Systems (FMS's), Aarhus University (2009)
3. Haage, I., Bastiaans, L., Kempenaar, C.: Exploring options for improved low dose application based on the MLHD-technology. In: Proceedings of the 12th EWRS Symposium Wageningen 2002, pp. 200–201 (2002)
4. Jørgensen, L., Kudsk, P.: Twenty years' experience with reduced agrochemical inputs: effect on farm economics, water quality, biodiversity and environment. In: HGCA Conference, Arable Crop Protection in the Balance: Profitand the Environment, pp. 1–10. HGCA, Lincolnshire, UK (2006)
5. Jørgensen, L.N., Noe, E., Langvad, A.M., Jensen, J., Ørum, J.E., Rydahl, P.: Decision support systems: barriers and farmers' need for support*. Bull. OEPP/EPPO Bull. **37**(2), 374–377 (2007)
6. Kempenaar, C., Groeneveld, R., Uffing, A., Van Der Weide, R., et al.: New insights and developments in the MLHD-concept of weed control. In: Symposium- European Weed Research Society, pp. 98–99 (2002)

7. Kudsk, P.: Experiences with reduced herbicide doses in Denmark and the development of the concept of factor-adjusted doses. In: Brighton Crop Protection Conference - Weeds, pp. 545–554 (1989)
8. Kudsk, P.: Optimising herbicide use-the driving force behind the development of the Danish decision support system. In: G. Marshal (ed.) Brighton Crop Protection Conference -Weeds, pp. 737–746 (1999)
9. Kudsk, P.: Optimising herbicide dose: a straightforward approach to reduce the risk of side effects of herbicides. Environmentalist **28**(1), 49–55 (2008)
10. Lancashire, P.D., Bleiholder, H., Boom, T., Langelüddeke, P., Stauss, R., WEBER, E., Witzenberger, A.: A uniform decimal code for growth stages of crops and weeds. Ann. Appl. Biol. **119**(3), 561–601 (1991)
11. Montull, J., Soenderskov, M., Rydahl, P., Boejer, O., Taberner, A.: Four years validation of decision support optimising herbicide dose in cereals under Spanish conditions. Crop Prot. **64**, 110–114 (2014)
12. Netland, J., Torresen, K., Rydahl, P.: Evaluation of the weed module in the danish decision support system "crop protection online" adapted to Norwegian conditions. In: Proceedings of the 13th EWRS Symposium, pp. 19–23 (2005)
13. Oerke, E.C.: Crop losses to pests. J. Agric. Sci. **144**(01), 31–43 (2006)
14. Rydahl, P.: A danish decision support system for integrated management of weeds. Asp. Appl. Biol. 43–53 (2004)
15. Seefeldt, S.S., Jensen, J.E., Fuerst, E.P.: Log-logistic analysis of herbicide dose-response relationships. Weed Technol. **9**, 218–227 (1995)
16. Statistics Denmark: http://www.statistikbanken.dk/statbank5a/selectvarval/saveselections.asp (2016). [Online]. Accessed 20 May 2016
17. Streibig, J.C., Kudsk, P., Jensen, J.E.: A general joint action model for herbicide mixtures. Pestic. Sci. **53**(1), 21–28 (1998)
18. Sønderskov, M., Fritzsche, R., de Mol, F., Gerowitt, B., Goltermann, S., Kierzek, R., Krawczyk, R., Bøjer, O.M., Rydahl, P.: DSSHerbicide: weed control in winter wheat with a decision support system in three South Baltic regions–field experimental results. Crop Prot. **76**, 15–23 (2015)
19. Sønderskov, M., Kudsk, P., Mathiassen, S.K., Bøjer, O.M., Rydahl, P.: Decision support system for optimized herbicide dose in spring barley. Weed Technol. **28**(1), 19–27 (2014)

Index

© Springer International Publishing Switzerland 2016

J. Papathanasiou et al. (eds.), *Real-World Decision Support Systems*,
Integrated Series in Information Systems 37,
DOI 10.1007/978-3-319-43916-7

U

V

W