Decision Engineering

Jatinder N.D. Gupta
Guisseppi A. Forgionne
Manuel Mora T.
Editors

Intelligent Decision-making Support Systems

Foundations, Applications and Challenges

Springer

Decision Engineering

Series Editor

Dr Rajkumar Roy
Department of Enterprise Integration
School of Industrial and Manufacturing Science
Cranfield University
Cranfield
Bedford
MK43 0AL
UK

Other titles published in this series

Cost Engineering in Practice
John McIlwraith

IPA – Concepts and Applications in Engineering
Jerzy Pokojski

Strategic Decision Making
Navneet Bhushan and Kanwal Rai

Product Lifecycle Management
John Stark

From Product Description to Cost: A Practical Approach
Volume 1: The Parametric Approach
Pierre Foussier

From Product Description to Cost: A Practical Approach
Volume 2: Building a Specific Model
Pierre Foussier

Decision-Making in Engineering Design
Yotaro Hatamura

Composite Systems Decisions
Mark Sh. Levin

Metaheuristics: A comprehensive guide to the design and implementation of effective optimisation strategies
Christian Prins, Marc Sevaux and Kenneth Sörensen
Publication due December 2006

Context-aware Emotion-based Multi-agent Systems
Rajiv Khosla, Nadia Bianchi-Berthouse, Mel Seigel and Toyoaki Nishida
Publication due February 2007

Jatinder N.D. Gupta, Guisseppi A. Forgionne
and Manuel Mora T.

Intelligent Decision-making Support Systems

Foundations, Applications and Challenges

With 105 Figures

Springer

Jatinder N.D. Gupta, PhD
College of Administrative Science
The University of Alabama in Huntsville
Huntsville, AL 35899
USA

Guisseppi A. Forgionne, PhD
Information Systems Department
University of Maryland Baltimore County
Catonsville, MD 21250
USA

Manuel Mora T. , EngD
Department of Information Systems
Autonomous University of Aguascalientes
Ave. Universidad 940
Aguascalientes
Ags. México 20100

British Library Cataloguing in Publication Data
Gupta, Jatinder N. D.
Intelligent decision-making support systems : foundations, applications and challenges. - (Decision engineering)
1.Decision support systems 2.Artificial intelligence
I.Title II.Forgionne, Guisseppi A., 1945- III.Mora, Manuel, 1961-
658.4'032
ISBN-10: 1846282284

Library of Congress Control Number: 2006922556

Decision Engineering Series ISSN 1619-5736
ISBN-10: 1-84628-228-4 e-ISBN 1-84628-231-4 Printed on acid-free paper
ISBN-13: 978-1-84628-228-7

Printed in Germany

9 8 7 6 5 4 3 2 1

Springer Science+Business Media
springer.com

Dedication

In memorial of the eminent scientist and
professor Herbert A. Simon (1916–2001)
for his outstanding contributions to the decision-making
systems and artificial intelligence research.

Jatinder N.D. Gupta, University of Alabama in Huntsville, AL, USA
Guisseppi A. Forgionne, University of Maryland Baltimore County, MD, USA
Manuel Mora, Autonomous University of Aguascalientes, AGS, México

Preface

Decision-making Support systems (DMSS) are *Information Systems* designed to interactively support all phases of a user's decision-making process. There are various notions about all aspects of this definition. There can be individual, group, and other users. Support can be direct or indirect. The decision-making process can be viewed in various ways. User-computer interaction can have a variety of dimensions. The information system offering the support can involve many technologies drawn from several disciplines, including accounting, cognitive science, computer science, economics, engineering, management science, and statistics, among others.

Research on DMSS can be focused on both organizational and technical issues. From a technical perspective, advances in *Information Technologies* (IT) have improved the support capabilities for DMSS. In particular, *Artificial Intelligence* (AI) has been recognized as a significant enhancement tool for DMSS. Despite the relevant research progress in both DMSS and AI, the majority of publications have been highly focused on either organizational or technical perspectives. While being effective and beneficial, this disparity has created much confusion about the theoretical basis, architectural forms, support mechanisms, design and development strategies, evaluation approaches, and managerial and organizational aspects of intelligent decision-making support systems (i-DMSS). This book, which we have titled, *Intelligent Decision-making Support Systems (i-DMSS): Foundations, Applications and Challenges*, is an attempt to alleviate some of the confusion.

Thus, this book aims to summarize and organize the main findings in both the DMSS and AI fields and focus the discussion on an integration of the tools for effective decision-making support. The book's mission is to present the core and state-of-the-art knowledge about intelligent decision-making support systems (i-DMSS). In the process, we hope to: (a) generate a compendium of quality theoretical and applied contributions in intelligent decision-making support systems (i-DMSS); (ii) disseminate scarce knowledge about foundations, architectures and effective and efficient methods and strategies for successfully designing, developing, implementing, and evaluating *intelligent decision-making support systems*, and (iii) create a bridge between DMSS and AI academicians and practitioners by promoting an awareness of the relevance of intelligent decision-making support systems in the current complex and dynamic management environment.

The presentation is divided into three sections. In the first section, labeled ***Foundations***, we provide theories, models or frameworks for i-DMSS, real or potential architectures for i-DMSS, comparative analyses among i-DMSS and other DMSS, and foundations of specific AI-based technologies and their relationships to i-DMSS. There are seven chapters in this first section.

In Chapter 1, the concept of intelligence in general, and artificial intelligence in particular, as it relates to aiding decision making is explored. The chapter then proposes an architecture for the evaluation of i-DMSS, and applies the model to empirical systems. The results are: (1) recognition of the contribution of AI to i-DMSS; (2) identification of the criterion (or criteria) used to evaluate i-DMSS; (3) categorization of the evaluation measures; (4) an architecture for the evaluation for i-DMSS; and (5) recommendation of a multicriteria model to assess i-DMSS.

Chapter 2 examines the legacy of Herbert Simon. Although his work has been used as the foundation for much research in i-DMSS, it is in the area of artificial intelligence that he has most directly contributed to the progress of scientific thinking. This chapter illustrates the impact of his research on i-DMSS and AI.

In Chapter 3, the authors reflect on the patterns of progress over the last ten years, compare them with the propositions put forth in 1992, and offer new perspectives based on an emerging stream in the literature base: intelligent inter-organizational decision support. The chapter looks to patterns in both the maturing streams of individual, team, and organizational intelligent decision support, as well as those emerging in the new interorganizational stream, to propose a direction for and identify significant challenges to be addressed in future work as intelligent decision support research enters its third decade.

The next chapter of this section, Chapter 4, investigates the combination of knowledge discovery in database and intelligent computing technologies, in developing a framework for intelligent decision-making support systems (i-DMSS). In this context, the chapter presents an approach for i-DMSS through the combination of data mining (DM) technology with artificial neural networks (NN) in a hybrid architecture called the DM-NN model. This research draws from the concepts of computational intelligence, knowledge discovery in databases and decision support.

Chapter 5 traces the development of AI and DMSS from their common origin in the ideas of Herbert Simon to the present time. It demonstrates that while AI has departed from those ideas, DMSS has remained largely influenced by them. Following a top-down approach, the chapter examines some of the basic premises of current DMSS to develop a new conceptual framework. The authors draw upon recent trends in AI towards situated models to propose an embedded, action-oriented, and improvizational approach to the design of intelligent decision-making support system (i-DMSS), and outline a methodology that would support this framework.

In Chapter 6, the authors review nine DMSS development processes and methodologies according to their focus on organizational issues, technical issues, or both. The chapter then proposes an innovative framework for the design and development of DMSS with particular attention on knowledge-management issues and their relationship with organizational and technical issues. It finally revisits two existing development processes according to this framework.

The last chapter of this section, Chapter 7, provides a framework for understanding the explanatory power of intelligent systems. It looks at content-based enhancements, drawn primarily from the expert systems literature, interface-based enhancements, and the appropriate selection of an advisory strategy. Such enhancements contribute to explanatory power by increasing system transparency and flexibility, and lead to outcomes such as better decision-making and problem-solving performance. Three illustrative examples demonstrate each type of enhancement of explanatory power. In the first case, a graphical hierarchy of an expert system knowledge base is illustrated. In the second case, the use of restrictive vs. nonrestrictive advisory strategies is discussed. Finally, deep explanations that provide a better understanding of a domain of expertise are described.

The second section of the book is called ***Applications***. As the label indicates, this section reports case studies of innovative real i-DMSS applications that deploy specific AI-based technologies, such as logic rule-based systems, neural networks, fuzzy logic, case-based reasoning, genetic algorithms, data-mining algorithms, intelligent agents, and user intelligent interfaces. Ten chapters detail these new i-DMSS applications.

In Chapter 8**,** the authors present evidence from Web-based i-DMSS, like comparison-shopping agents, that show distinctive patterns of information systems development (ISD). Since there are few studies that cover the ISD paradigm for agents in an open-domain system, through three mini case studies in this chapter, the authors provide relatively comprehensive cases for future theory formulation.

Chapter 9 proposes a new negotiation-support mechanism that can be utilized to incorporate causal relationships between structured negotiation terms (SNT) and unstructured negotiation terms (UNT) in the process of B2B negotiation, by using a cognitive map. The proposed negotiation mechanism suggests that cognitive maps could be used to represent causal relationships between SNTs and UNTs, both as knowledge-representation vehicles and as inference engines. After reviewing the potential of cognitive map in B2B negotiation, a prototype, CAKES-NEGO, is presented, which is then used, through illustrative examples, to examine the validity of the proposed mechanism. Statistical tests indicated that the proposed negotiation mechanism could improve decision performance significantly in B2B negotiations.

Chapter 10 describes how "information overload" often results in wasted time and resources and inefficient and unproductive knowledge discovery. In theory, the concept of *Just in Time Knowledge Management* (JITKM) can help resolve this problem. This chapter tests the theory empirically, and this test supports the theory.

In Chapter 11, the authors present the development of an intelligent decision-making support system (i-DMSS) for regional aquaculture planning. The i-DMSS applies fuzzy set theory to multiple criteria decision-making (MCDM) in regional aquaculture planning. A case study from Egypt demonstrates the proposed IDSS and the fuzzy MCDM framework.

According to the authors of Chapter 12, practice has shown that allocation and routing decisions made manually by human operators with long experience are usually nearly optimal, and it is very hard to beat those decisions using a computerized DMSS. They present an i-DMSS that can help the new decision maker to reach decisions comparable in quality to those made by a retiring pair of senior decision makers in a bus rental company in Seoul, South Korea. In this chapter, the authors discuss this decision problem, its context, the models used to solve it, the algorithms used in the i-DMSS to solve these models, and how this i-DMSS is used to make the decisions daily. They report that the i-DMSS is based on bipartite matching and transportation algorithms and heuristics, and produces solutions 10-20% more economical than the manual decisions.

The recent massive use of wireless technology in the business domain strongly modified organization and management of work and made it critical both to gather decision-problem data and to share them among human-business agents in real-time. To support this new generation of decision makers and face real-time in-the field decision problems, the author of Chapter 13 developed MicroDEMON, a language-based user-centered mobile software system that represents a step further along the evolution of *Active* DMSSs, a kind of intelligent and proactive decision-making support system.

Chapter 14 presents an intelligent decision support system (i-DMSS) that helps decision makers to identify key issues and to improve policy-making processes, particularly with respect to strategic decisions. The i-DMSS combines artificial intelligence (AI) techniques with qualitative models and system dynamics simulation, and allows quantitative and qualitative variables to be integrated into a comprehensive methodology, necessary to design strategies and policies. The results of this study are currently in the process of being implemented.

In Chapter 15, the authors review electronic negotiation systems and intelligent agents for negotiations. Different types of negotiation agents, their roles and requirements, and various methods for effective support or conduct of negotiations are discussed. Selected applications of intelligent negotiation agents are presented.

Chapter 16 describes a hybrid decision model and a multiagent framework for collaborative decision support in the design process. The proposed knowledge-based collaborative decision support model can quantitatively incorporate qualitative design knowledge and preferences for multiple, conflicting attributes stored in a knowledge repository so that a better understanding of the consequences of design decisions can be achieved from an overall perspective. The multiagent framework provides an efficient decision support environment involving distributed resources to shorten the realization of products with optimal life-cycle performance and competitiveness. This framework is illustrated with an application in concept evaluation and selection in power supply product family design for mass customization.

The last chapter of this section, Chapter 17, examines the potential of using Semantic Web technologies as part of an intelligent decision support system (IDSS). The Semantic Web is introduced and its benefits to i-DMSS are highlighted. In addition, two Semantic Web ontologies for i-DMSS are developed using the Protégé tool. The two ontologies are used to demonstrate their ability to infer new knowledge, help visualize and improve data presentation, query data and allow global database linkage as a part of the Semantic Web. These characteristics allow the ontologies to be processed by intelligent, web-based software agents.

The third section of the book is titled ***Trends and Challenges***. This section presents chapters that analyze the implications, challenges and trends of i-DMSS for individual, team, organizational or interorganizational decision-making processes, from a technical and organizational perspective. These challenges and trends developments are offered in six separate chapters.

In Chapter 18, the author addresses complexities of inference-based decision making and the challenge of supporting such processes. People are often unaware of the cognitive framework that facilitates and biases complex decisions. Inference-based decision making is described as bidirectional reasoning. The chapter shows how the decision maker gradually makes sense of and simplifies the decision task, how decision criteria can be modeled, and how criteria change as the decision-maker's experience increases. It is claimed that the prime way to aid inference-based decision making is to make the process of generating sense explicit, and that experiments constitute an important tool.

Chapter 19 reviews the knowledge-based view on decision support and argues for the emergence of a new type of intelligent decision-making support system – an intelligent gateway for supporting specific knowledge needs. The modern view on decision support and expert systems has shifted from considering these as purely analytical tools for assessing best decision options to seeing them as a more comprehensive environment for supporting efficient information processing based on a good understanding of the problem context. Such intelligent decision support systems incorporate problem domain knowledge to improve their information processing and provision capabilities. More recently, information portals have been proposed as tools for matching users' information needs in order to enhance their decision-making ability. This chapter looks at portals as new types of intelligent decision support systems, which use problem-domain knowledge in order to improve efficiency in information provision. The main focus of the chapter is on suggesting mechanisms for implementing intelligent decision support capabilities in a healthcare portal, which seeks to deliver personalized information to support efficient decision making. BCKOnline, a healthcare portal built around breast cancer information, is described as an example of such implementation.

Arguing that contemporary decision making needs to be tackled through a holistic perspective, in that the conceptual, methodological and application-oriented aspects of the problem have to be simultaneously taken into account, Chapter 20 provides an overview of challenges for the future development of decision support technologies and their integration in intelligent decision-making support systems. Based on this discussion, and aiming at providing decision makers around the world with applications of enhanced performance, while, at the same time, addressing their communication and collaboration needs in an efficient and effective way, an advanced Web-based decision-making framework is proposed.

Chapter 21 reports the findings of exploratory research into the technical and organizational challenges facing i-DMSS for nonoperational decision making and indicates directions for future research. Some challenges arise as a direct result of expert judgments used to compensate for the lack of information inherent in complex, uncertain situations, and the necessity to support intelligently the interaction between decision makers and experts, each of them having different backgrounds, interests, knowledge and cognitive style. Another important challenge results from the manipulation and interpretation of information, especially for expert judgments' aggregation and provision of advanced functionalities such as forecasting in the context of highly uncertain and complex decisions.

In Chapter 22, the authors describe a new software product called the Planners Lab©. The software is a DSS -in this case an abbreviation for *Decision Support Simulator*-. The software is intended to be used as a laboratory for academic teaching, research and consulting. This chapter describes the current state of the software, giving inferences to how this software can improve decision making. Several potential research streams are described as well.

Chapter 23 examines past and current i-DMSS literature. This examination is used to identify patterns, trends, and remaining challenges. The information is used to suggest a strategic research agenda for the i-DMSS field.

The final chapter in the book, Chapter 24, traces the impact of decision support methods, including those based on *Artificial Intelligence* concepts, from the beginning, through the present, and concludes with proposals for the future of the profession. The chapter outlines the history of the DSS concept from the start, then reviews the problems currently being addressed, before moving on to consider the global scale of the challenges that lie ahead.

We believe that the book will be a comprehensive compilation of i-DMSS thought and vision. There is a thorough presentation on all phases of decision-making support, newly reported applications in i-DMSS in a variety of areas, unique information technologies for improving i-DMSS design, development, and implementation, unique strategies for measuring i-DMSS effectiveness, and new methodologies for managing i-DMSS in practice. The presentation illustrates the concepts with a variety of public, private, societal, and organizational applications, offers practical guidelines for designing, developing, and implementing i-DMSS, offers measures to effectively evaluate and manage i-DMSS, and presents expert opinion about the future of i-DMSS.

Readers of the text will gain an understanding of, among other things: (a) decision-making concepts in organizations, (b) i-DMSS types, (c) i-DMSS integration strategies, (d) i-DMSS architectures, (e) i-DMSS design, development, and implementation strategies, tools, and methodologies, (f) i-DMSS implementation barriers, and (g) future i-DMSS trends. Such knowledge will facilitate the development and implementation of intelligent decision-making support systems within any organization. It is hoped that the book will enable the business community to start benefiting more widely from this powerful technology.

This understanding of various phases of i-DMSS should benefit graduate students taking decision-making support systems courses and practitioners seeking to better support and improve their decision making. Hopefully, the book will also stimulate new research in i-DMSS by academicians and practitioners.

Acknowledgments

This book would not have been possible without the cooperation of many people: the authors, reviewers, our international colleagues, and the staff at Springer-Verlag. The editors would like to thank Anthony Doyle for his academic endorsement to produce this book and Kate Brown as development editor for her timely responses to our questions and keeping us on schedule. The resources that the staff at Springer-Verlag provided assisted us enormously and specially to Sorina Moosdorf by her gentle indications.

Many of the authors of chapters in this book also served as reviewers of other chapters, and so we are doubly appreciative of their contribution. We also acknowledge our respective universities for affording us the time to work on this project and international colleagues for many stimulating discussions.

Finally, the authors wish to acknowledge their families for providing time and support for this project.

Jatinder N.D. Gupta, University of Alabama in Huntsville, AL, USA
Guisseppi A. Forgionne, University of Maryland Baltimore County, MD, USA
Manuel Mora, Autonomous University of Aguascalientes, AGS, México

Table of Contents

Part II Applications of i-DMSS

Part III Trends of i-DMSS

List of Contributors

F. Adam
Business Information Systems
University College Cork
Cork, Ireland
fadam@afis.ucc.ie

P. Braun
Faculty of Information and
Communication Technologies
Swinburne
University of Technology
Hawthorn, Australia
pbraun@it.swin.edu.au

J. Brzostowski
Faculty of Information and
Communication Technologies
Swinburne
University of Technology,
Hawthorn, Australia
jbrzostowski@it.swin.edu.au

F. Burstein
School of Information Management
and Systems, Monash University,
Caulfield Campus, 26 Sir John
Monash Drive, Caulfield East,
Victoria, Australia.
frada.burstein@
infotech.monash.ed.au.

N. Cassaigne
The University of Manchester
Manchester, United Kingdom
nathalie.cassaigne@
manchester.ac.uk

F. Cervantes
Center of Applied Sciences and
Technology Development,
National Autonomous University of
Mexico (UNAM),
Mexico, D.F. 04510
francisco_cervantes@
cuaed.unam.mx

N. Conteh
Former PhD student in:
Department of Information Systems
College of Information Technology
and Engineering
University of Maryland Baltimore
County (UMBC)
Baltimore, MD 21250, USA
nconteh1@umbc.edu

B. Correa-Pérez
Florida International University
Miami, Florida 33199 USA
bertha.correa@fiu.edu

J. Courtney
College of Business,
University of Central Florida,
Orlando, FL, USA

H. Ekbia
Computer Science Department and
The Redlands Institute
University of Redlands,
1200 E. Colton Ave. Redlands,
CA 92373, USA
hamid_ekbia@redlands.edu

O.F. El-Gayar
College of Business and Information
Systems
Dakota State University
Madison South Dakota, USA
Omar.el-gayar@dsu.edu,

J. Fisher
Enterprise Information Research
Group
Monash University,
Melbourne, Australia
PO Box 197, Caulfield East, 3145,
Victoria, Australia
Julie.Fisher@sims.monash.edu.au

G.A. Forgionne
Department of Information Systems
University of Maryland Baltimore
County
Catonsville, MD 21250, USA
forgionn@umbc.edu

A. Gachet
Department of Information
Technology Management
University of Hawaii at Manoa
2404 Maile Way, A202
Honolulu, HI 96822, USA
gachet@hawaii.edu

L. Garrido
Center of Intelligent Systems
Monterrey Tech (ITESM)
Monterrey, N.L., Mexico, 64800
leonardo.garrido@itesm.mx

O. Gelman
Center of Applied Sciences and
Technology Development
National Autonomous University of
Mexico (UNAM)
Mexico, D.F. 04510
ogelman@servidor.unam.mx

C. González
School of Computer Science
University of La Laguna
38071, La Laguna, Spain.
cjgonza@ull.es

M. Goul
Department of Information Systems
W.P. Carey School of Business
Arizona State University
Tempe, AZ, USA
Michael.Goul@asu.edu

J.N.D. Gupta
College of Administrative Science
The University of Alabama
in Huntsville
Huntsville, AL 35899, USA
guptaj@uah.edu

P. Haettenschwiler
Department of Informatics
University of Fribourg
Faucigny 2, 1700
Fribourg, Switzerland
pius.haettenschwiler@unifr.ch

N. Karacapilidis
Industrial Management and
Information Systems Lab MEAD
University of Patras
26504 Rio Patras, Greece
nikos@mech.upatras.gr

G. Kersten
John Molson School of Business
Concordia University
Montreal, Canada
kersten@jmsb.concordia.ca

J. Kim
John Molson School of Business
Concordia University
Montreal, Canada
jbkim@jmsb.concordia.ca

W. Kim
Department of Industrial and
Information Systems Engineering
Seoul National University of
Technology
172 Gongneung 2-dong
Nowon-gu, Seoul 139-743
South Korea
wjkim@snut.ac.kr

R. Kowalczyk
Faculty of Information and
Communication Technologies
Swinburne, University of
Technology, Hawthorn, Australia
rkowalczyk@it.swin.edu.au

S. Kwon
School of Business Administration
Sungkyunkwan University
Seoul 110-745, Korea
kwonsj@skku.edu

K. Lang
Department of Computer
Information Systems
Baruch College
City University of New York
New York, 10010, USA
karl_lang@baruch.cuny.edu

K.C. Lee
School of Business Administration
Sungkyunkwan University
Seoul 110-745, South Korea
leekc@skku.ac.kr

C. Legna
Department of Institutional
Economics,
University of La Laguna,
38071, La Laguna, Spain.
clegna@ull.es

L. Lorimier
PhD student in:
The University of Manchester
Manchester, United Kingdom
lionel_lorimier@hotmail.com

J. Lu
School of Engineering
The University of Birmingham
Birmingham, B15 2TT, UK
j.lu@bham.ac.uk

C. Lundberg
A.J. Palumbo School of Business
Administration & John F. Donahue
Graduate School of Business
Duquesne University,
600 Forbes Avenue,
Pittsburgh, PA 15282, USA &
Swedish School of Economics and
Business Administration,
Helsinki, Finland
lundberg@duq.edu

K. Madelin
School of Engineering
The University of Birmingham
Birmingham, B15 2TT, UK
j.lu@bham.ac.uk

R. Manaszewicz
Enterprise Information Research Group
Monash University
Melbourne, Australia
PO Box 197, Caulfield East, 3145
Victoria, Australia

P. Malhotra
Enterprise Information Research Group
Monash University
Melbourne, Australia
PO Box 197, Caulfield East, 3145
Victoria, Australia

A. McCosh
Florida International University
Miami, Florida 33199 USA
mccosha@fiu.edu

S. Mckemmish
Enterprise Information Research Group
Monash University,
Melbourne, Australia :
PO Box 197, Caulfield East, 3145
Victoria, Australia
Sue.McKemmish@sims.monash.edu.au

M. Mora
Department of Information Systems
Autonomous University of Aguascalientes
Aguascalientes, Mexico, 20100
mmora@securenym.net

K. Murty
Department of Industrial and Operations Engineering
University of Michigan
Ann Arbor, Michigan 48109-2117
USA
murty@umich.edu

R. Nakatsu
College of Business Administration
Department of Computer Information Systems,
Loyola Marymount University
1 LMU Drive,
Los Angeles, CA 90045, USA
makatsu@lmu.edu

J. Nichols
PhD Student in:
Department of Information Systems
W.P. Carey School of Business
Arizona State University
Tempe, AZ USA
Jason.Nichols@asu.edu

D. Paradice
College of Business,
Florida State University
Tallahassee, FL, USA

G. Phillips-Wren
Department of Information Systems
Loyola College in Maryland
Baltimore, MD 2121, USA
gwren@loyola.edu

G. Pistolesi
ThinkinGolem SC
Piazza Prati degli Strozzi 21,00195
Rome, Italy
guliano.pistolesi@thinkingolem.com

J.C. Pomerol
Université Pierre et Marie Curie
4 Place Jussieu, 75252
Paris 5, France
Jean-Charles.Pomerol@lip6.fr

C. Roberts
School of Engineering
The University of Birmingham
Birmingham, B15 2TT, UK
j.lu@bham.ac.uk

A. Stirling
School of Engineering
The University of Birmingham
Birmingham, B15 2TT, UK
j.lu@bham.ac.uk

R.D. Sriram
Manufacturing System Integration Division,
National Institute of Standards and Technology,
Gaithersburg, MD 20899, USA

S. Strecker
John Molson School of Business
Concordia University
Montreal, Canada
strecker @jmsb.concordia.ca

K. Tandekar
College of Business and Information Systems
Dakota State University
Madison, South Dakota, USA
tandek@pluto.dsu.edu

R. Vahidov
John Molson School of Business
Concordia University
Montreal, Canada
vahidov@jmsb.concordia.ca

S. Viademonte
School of Information Management & Systems,
Monash University
Caulfield Campus
26 Sir John Monash Drive
Caulfield East, Victoria, Australia.
sergio.viademonte
@infotech.monash.edu.au.

G. Wagner
Peter Kiewit Institute
University of Nebraska
Omaha, NE, USA

Y. Wan
Department of Computer Science
University of Houston, Victoria
4902 Mulberry Circle
Missouri City, TX 77459, USA
wany@uhv.edu

X.F. Zha
Manufacturing System Integration Division,
National Institute of Standards and Technology,
Gaithersburg, MD 20899, USA
zha@cme.nist.gov

Part I

Foundations of Intelligent Decision-making Support Systems

1

A Multicriteria Model for the Evaluation of Intelligent Decision-making Support Systems (i-DMSS)

Gloria Phillips-Wren[1], Manuel Mora[2], Guisseppi A. Forgionne[3], Leonardo Garrido[4] and Jatinder N.D. Gupta[5]

[1] Department of Information Systems, Loyola College in Maryland, Baltimore, MD 21210, USA, e-mail: gwren@loyola.edu
[2] Department of Information Systems, Autonomous University of Aguascalientes, Aguascalientes, Mexico, 20000, e-mail: mmora@securenym.net
[3] Department of Information Systems, University of Maryland Baltimore County, Catonsville, MD 21250 USA, e-mail: forgionn@umbc.edu
[4] Center for Intelligent Systems, Monterrey Tech, Monterrey, N.L., Mexico, 64800, e-mail: leonardo.garrido@itesm.mx
[5] College of Administrative Science, The University of Alabama in Huntsville, Huntsville, AL 35899, USA, e-mail: guptaj@uah.edu

Although traditional decision-making support systems (DMSS) have been researched extensively, few, if any, studies have addressed a unifying architecture for the evaluation of intelligent DMSS (i-DMSS). Traditional systems have often been evaluated in the literature on the basis of single-outcome measures, such as decreased cost, increased profit, or improved forecasting, compared to decision making without a DMSS. In cases in which other metrics are used for evaluation, process measures are most often cited, such as increased efficiency, organizational learning, and increased speed. Previous research by the authors has shown that a multicriteria evaluation for DMSS can be provided, combining both outcome and process measures into a single metric using the analytic hierarchy process (AHP). However, the specific categories that should be utilized as evaluation measures have not been defined, and no studies have focused exclusively on categories for the evaluation of i-DMSS. This chapter explores the concept of intelligence in general, and artificial intelligence in particular, as it relates to aiding decision making. It then proposes an architecture for the evaluation of i-DMSS and applies the model to empirical systems. The results are: (1) recognition of the contribution of AI to i-DMSS; (2) identification of the criterion (or criteria) used to evaluate i-DMSS; (3) categorization of the evaluation measures; (4) an architecture for evaluation for i-DMSS; and (5) recommendation of a multicriteria model to assess i-DMSS.

1.1 Introduction

Decision makers are faced with increasingly stressful environments – highly competitive, fast-paced, near real-time, overloaded with information, data distributed throughout the enterprise, and multinational in scope. The combination of the Internet enabling speed and access, and the maturation of artificial intelligence techniques, has led to sophisticated aids to support decision making under these risky and uncertain conditions. These aids have the potential to improve decision making by suggesting solutions that are better than those made by the human alone. They are increasingly available in diverse fields from medical diagnosis to traffic control to engineering applications (see Table 1.7). We have called these aids intelligent decision-making support systems (i-DMSS), and we begin this chapter with an exploration of intelligence as it applies to decision aids. These systems can significantly affect both the process of, and outcome from, decision making. We thus propose that both factors should be considered in their evaluation.

In a study of traditional decision support systems (DSS), Forgionne (1999) found that most empirical studies focused on either process-oriented or outcome-oriented evaluation measures, but not both in the same study. In addition, quantitative outcome measures were most often used for evaluation, such as increased profit or decreased cost. When multiple measures were used, outcome and process-oriented measures were presented as individual values with no integrative assessment of the overall value of the DSS. In the case of i-DMSS, a consideration of multiple evaluation criteria is particularly relevant. Not only is the outcome of the decision affected by using intelligent techniques, but with the increasing use of Web-based and real-time DSSs features, the process of decision-making is affected as well (Grabowski and Sanborn 2001). Such systems enhance and extend traditional DSSs by providing features such as just-in-time information, real-time processing, online transaction processing, connectivity and globally up-to-date information (Phillips-Wren and Forgionne 2002).

Although traditional DMSS have been researched extensively, few, if any, studies have addressed a unifying architecture for the evaluation of i-DMSS. The necessity of a multicriteria DMSS evaluation has been reported in the literature. For example, Maynard *et al.* (2001) suggested a multicriteria approach to include the perspectives of different constituencies or stakeholders. Adelman (1992) proposed a multifaceted evaluation approach for DMSS and expert systems consisting of subjective, technical and empirical evaluation methods. The subjective methods were proposed to assess the system from the perspective of users and sponsors; the technical evaluation method focused on the analytic methods used in the DMSS; and the empirical methods focused on a comparison of performance with and without the system. The current proposed approach extends previous studies by suggesting a unifying architecture for evaluation using the analytic hierarchy process (AHP), while focusing specifically on the support of decision making.

Previous research (Forgionne 1999; Phillips-Wren and Forgionne 2002; Phillips-Wren *et al.* 2004) has shown that a multi-criteria evaluation for i-DMSS can be provided using the AHP considering both outcome and process measures to provide a single metric. However, a study of the specific categories that should be utilized under the AHP metric have not been defined, and no studies have focused exclusively on i-DMSS. Consequently, this chapter assesses empirical i-DMSS from the literature with the goals of:

(1) identifying the criterion (or criteria) used to evaluate i-DMSS in the literature;
(2) categorizing the evaluation measures;
(3) suggesting appropriate evaluation categories for i-DMSS; and
(4) recommending a multicriteria model to assess i-DMSS.

It must be noted that the implicit decision to be made in the AHP-based model is the selection of the best design of several i-DMSS candidate architectures based on the ranks assigned by the process and outcome contributions. This approach is a postmortem evaluation, but with further modifications to predict impacts, it can be used as a priori evaluation method. Consequently, the posed assessment conceptual tool is expected to offer guidance for a better architectural design and evaluation of i-DMSS by accounting for a system's AI-based features within the process and outcomes assessment. Such an approach links AI investments based in improvements in computer velocity and memory gains (the fundamental Computer Science perspective) to measurable individual, team or organizational performance.

The chapter is organized in the following manner. Section 1.2 provides an overview of the generic decision-making process and an examination of the concept of intelligent systems from the main AI and DMSS literature. Section 1.3 reviews a number of empirical studies of i-DMSS to elucidate the metrics that are used in the literature for evaluation. Section 1.4 presents the results of a multicriteria evaluation schema based on the empirical studies. Finally, section 1.5 gives the conclusions and contributions of this research

1.2 Theoretical Background and Related Work

In this section we first examine a generic decision-making process and the definitions of intelligent systems from the AI and DMSS literature to establish a common vocabulary and theoretical base. We then review the components of generic architectures posed for i-DMSS that incorporates complexity levels for the information system. Then, an integrated i-DMSS model is offered.

1.2.1 A Generic Decision-making Process

Nobel laureate Herbert A. Simon visualized the decision-making process as the search for feasible paths in a searching-space (Simon 1996, p. 86). This visualization suggests that decision making can be considered also as "*a process of choosing among alternative courses of action for the purpose of attaining a goal or goals*" (Turban and Aronson 1998, p 34). Several models have been reported in the literature on how the decision-making process should be, or is actually, conducted (Turban and Aronson, 1998, Forgionne, 1999, Mora *et al.* 2003) The most popular are based on Simon's Model of three phases (1997, first reported in "Administrative

Behavior", 1947) and recent extensions (Mora *et al.* 2003). Table 1.1 summarizes the extended five-phase model.

Table 1.1 Decision-Making Phases and Steps

PHASE	STEP	DESCRIPTION
Intelligence ("Simon's setting the agenda step")	Data Gathering	Observation of reality and collecting of any relevant qualitative and quantitative data is done for the general situation of interest.
	Problem Recognition	Based on the interpretation of collected data, a well-focused problem statement and general objective is defined.
Design (Simon's representing the problem step)	Model Formulation	Using the well-focused problem, a predefined model is instanced with a set of courses of action, outcomes criteria, set of uncontrolled events and parameters, and the relationships between these variables. If a predefined model is unavailable, a new model must be developed.
	Model Analysis	Face validity and pilot test of the model is conducted to reduce any potential source of significant error.
Choice (Simon's finding and selecting alternatives steps)	Generation & Evaluation	With a validated model, all courses of action are evaluated (or dynamically generated) and what-if, sensitivity, and goal-seeking analysis are conducted, in terms of the outcomes criteria.
	Selection	Best course of action is finally suggested, using an optimization, satisfaction criteria, or other approach.
Implementation	Result Presentation	Selected course of action is reported to top management team for final organizational authorization. (a decision can be taken but not implemented)
	Task Planning	Decision authorized, is scheduled in a set of specific actions, where financial, human and material resources are estimated.
	Task Tracking	The set of specific actions are conducted and monitored until the planned end action is achieved.
Learning	Outcome-process Analysis	Process and outcomes metrics are collected from decision-making team and organization.
	Outcome-process Synthesis	Learned lessons on the decision-making process are identified and communicated to the top management team.

The decision-making process is a complex task. It is continuous and partially iterative in that the phases may overlap, and the decision maker may loop back to a previous phase (Simon, 1997). However, although some steps may be performed concurrently, decision making is fundamentally a sequential process with "design" requiring "intelligence", "choice" needing "design", and "implementation" following "choice" (Forgionne 1999). These steps are repeated iteratively with many feedback loops until the final choice has been implemented and lessons learned have been identified and communicated.

1.2.2 Concept of Intelligent Systems from AI Literature

The concept of intelligence has been extensively debated in the psychology and related literature (Jensen 1999). Whereas a standard definition still remains elusive, there are some literature-based common characteristics of "human intelligence". Such intelligence involves a learning ability (*i. e.* to increase conceptual and procedural knowledge), understanding and communication of messages (*i. e.*, to make sense of messages and generate expected responses), making decisions and problem solving in a rational way, and developing new abstract and physical artifacts to cope with survival and development in society.

In the mid-1950's, the Nobel laureate Herbert A. Simon (1996) proffered that human behavior should be studied with other techniques beside the standard and dominant paradigm of Psychology. Simon (1996) and colleagues sought to expand the "empty organism" model, (Boering 1933) that merely correlated independent (*i. e.* stimulus) and dependent variables (*i. e.* responses), by studying explanatory mechanisms for the organism. In contrast with Boering's thesis, which claimed that explanatory mechanisms should be two-layered neurobiological, Simon and colleagues posed several layers. Simon noted that *"the information processing theories of cognition represent a specific* intermediate *layer of explanation*" (Simon, 1996, pp. 192), and Newell and Simon (1976) established the "physical symbol processing conjecture" In this view, the human mind is a natural symbol processing system, and the research efforts of AI can be basically focused on the design and testing of symbolic systems using the computer as the experimental site. In particular, Simon and Newell (ibid, pp. 114) suggest that the intelligence level of a system can be measured "*by its ability to achieve stated ends in the face of variations, difficulties and complexities posed by the task environment*". Implicitly, these researchers suggested that AI's long-term aim is the engineering of intelligent artifacts that mimic human beings.

Alan Turing (1950), in his seminal paper titled "Computing Machinery and Intelligence", established a test for the Simon and Newell concept. Basically, the Turing Test (TT) stated that a machine should be considered intelligent if a human being cannot distinguish the interaction (via two computer terminals) between the computer program and a human being using one of the terminals. In this sense, the computer program could be considered intelligent because the program could imitate rational and intelligent human behavior. In turn, John McCarthy (2003, p. 3) introduced the "heuristic hypothesis" of the psychologist A. Jensen (1998). According to this hypothesis, *"…all normal humans have the same intellectual mechanisms* [*i. e.*, "hardwiring"] *and that differences in intelligence are related to quantitative biochemical and physiological conditions. I see them as speed, short term memory, and the ability to form accurate and retrievable long term memories*". Jensen (1998) also supported the existence of a general highest-order common factor of intelligence, called the g Factor, to account for the specific biochemical and physiological conditions. Jensen agrees that intelligent actions (such as "*attention, perception, discrimination, generalization, learning, memory, language, thinking, problem-solving, and the like*") are information-processing tasks (Jensen 2000).

Using such a premise, McCarthy suggested that intelligent artifacts are based on the intellectual underlying mechanisms put forth by their designers (ibid, pp. 4).

This idea agrees with the need to study the layer of explanatory mechanisms for the "empty organism".

The real potential of building intelligent systems also has been studied by non-AI-based scientists. Parnas (1985, pp. 20-21) pointed out that AI should be focused on the design of new algorithms based on strong engineering and scientific processes. In this view, emulating human-like heuristics is difficult because of human limitations and the incompleteness of trial and error techniques used to develop rule-based systems. Simon and Newell (1976, p. 114) also defended the same premise: "*computer science* [referring to AI] *is an unempirical discipline. We would have called it an experimental science, but like astronomy, economics, and geology, some of its unique forms of observation and experience do not fit a narrow stereotype of the experimental method. None the less, they are experiment*". Brooks (1996, pp. 63:64) posed that "IA > AI"– intelligence amplifying systems are better than artificial intelligence systems. This view supports the notion of "weak AI", which holds that human minds must be studied to create better mechanisms for intelligent systems that will assist, but not replace, humans.

The weak AI notion supports the view that decision support systems should assist, but not replace, humans in the decision-making process (Eom *et al.* 1998). This notion is also consistent with Simon's premise of bounded rationality in humans and the concept of building better decision tools to overcome and amplify human-based capabilities.

1.2.3 Fusion of Decision Support Systems and AI Techniques: Intelligent Decision-making Support Systems (i-DMSS)

In general, decision support systems (DSSs) provide support for decision makers by bringing together human judgment and computerized information in an attempt to improve the effectiveness of decision-making (Turban and Aronson 1998, p. 77). The general purpose of a DSS can be stated as "to supplement one or more of a decision maker's abilities" (Holsapple and Whinston 1996, p. 136). Holsapple and Whinston (1996, p. 144-145), forerunners in the design and study of intelligent DSS, which were called knowledge-based decision support systems (KB-DSS), have suggested the following characteristics for such systems (second generation of DSS oriented to become current i-DMSS): (a) it contains various types of knowledge that describe selected aspects of the decision-maker's world; (b) it has an ability to acquire and maintain descriptive knowledge such as record keeping and other types of knowledge as well; (c) it can produce and present knowledge in various ways; (d) it can select knowledge to present or derive new knowledge; (e) it can interact directly [intelligently] with the decision maker.

A review of the main DMSS literature in the medical, military, financial, political, and environmental contexts helps to offer a combined perspective. Intelligent human-like support is needed for decision-making support, but human decision makers should make the final and critical decisions (Macintosh 2004). Table 1.2 summarizes the concept of "intelligent behavior" from the DMSS literature.

Table 1.2. Definitions of intelligent behavior in DMSS literature

Concept of human-like intelligent actions	Sources
Intelligent systems should be able to: (i) learn or understand from experience; (ii) make sense out of ambiguous or contradictory messages; (iii) respond quickly and successfully to a new situation; (iv) use reasoning in solving problems and directing conduct effectively; (v) deal with perplexing situations; (vi) understand and infer in ordinary, rational ways; (vii) apply knowledge to manipulate the environment; (viii) think and reason; and (ix) recognize the relative importance of different elements in a situation.	Turban and Aronson (1998, p. 199)
Intelligence as "acting as we would expect people to act" and state that artificial intelligence provides the techniques.	Brown and O'Leary (1995, p. 1).
Intelligence as *"the ability of a system to behave appropriately in an uncertain environment, where appropriate behavior is that which maximizes the likelihood of success in achieving the system's goals."* Their definition is intended to span a spectrum of capability from simple to complex and recognizes degrees of intelligence. They claim that the degree of intelligence is determined by three factors: (1) available computational power and memory; (2) sophistication of the underlying processes or models; and, (3) the quality and quantity of information and values available to the system.	Albus and Meystel (2001, p. 6)
'Intelligence' in the context of technology systems is not synonymous with 'human intelligence'. The system is referred to as intelligent if it exhibits some of the abilities that are associated with 'intelligent behavior'.	Pohl (2005)

An i-DMSS extends traditional DSS by incorporating techniques to supply intelligent behaviors and utilizing the power of modern computers to support and enhance decision making (Proudlove *et al.* 1998, Gottinger and Weimann 1992, Elam and Konsynski 1987). The i-DMSS may, for example, respond quickly and successfully to new data and information without human intervention, deal with perplexing and complex situations, learn from previous experience, apply knowledge to understand the environment, recognize the relative importance of different elements in the decision, incorporate the knowledge of domain experts, recommend action, and/or act on behalf of the human (by a predefined authorization of the decision-maker).

Some reports in the literature (Goul *et al.* 1992; Eom 1998) have highlighted the contributions of AI to the DSS discipline. However, a recent study (Mora *et al.* 2003) identified a crisis situation in the field. Despite excellent results with AI proof of concepts for the design of i-DMSS, the concepts have effectively supported only some decision-making steps and phases in the last 20 years. In the aforementioned study, a conceptual architecture for i-DMSS was developed with the purpose of assessing, from a top-level perspective, the underlying levels of capabilities available in the DMSS.

The conceptual capability assessment framework for DMSS (CAF-DMSS) has been updated from previous versions reported by same authors (Mora *et al.* 2003). Basically, the extensions involve the new dimension of user interface support and a consolidation of the previous processing levels. As a result, this new framework has three axes as core structural components: (i) user interface capabilities, (ii) data, information and knowledge representation, and (iii) processing support mechanisms.

The first and second dimensions are based in the general and standard structure for a DMSS (Sprague 1980, Turban and Aronson 1998). The third dimension is based on the types of decisions tasks, levels of intelligence embedded in algorithms and types of intelligent operations for intelligent data-mining systems suggested in the literature (Elam and Konsynski 1987, Dhar and Stein 1998, Gray and Watson 1996). A previous framework reported by authors and this one integrate the findings from the literature and establish a conceptual ordinal scale for the three dimensions. The first axis (user interface capabilities) is divided into three levels that examine the richness of the presentation and action language. Table 1.3 presents a description of these capability levels.

Table 1.3. Levels of user interface (UI) capability

Levels of UI Dimension	Description of capability
I	The DMSS provides an action language of structured commands and/or menus and a presentation language based on texts and non-dynamic or animated graphics.
II	The DMSS provides an action language of structured commands and/or menus and a presentation language based on hypertext, or multimedia graphics, sounds, animations and video or dynamic graphics or simulation-based outcomes.
III	The DMSS provides an action language of natural language and a presentation language based on virtual reality environments.

The data, information and knowledge representation axis is divided into five levels of structural complexity, as described in Table 1.4.

Table 1.4. Levels of data, information and knowledge (DI&K) representation capability

Levels of DI&K Dimension	Description of Capability
I	The DMSS uses plain files, simple data structures or/and one-dimensional database schemes to represent data and information items.
II	The DMSS uses complex and highly structured data structures or/and multidimensional database schemes to represent data and information items.
III	The DMSS accesses structured data, information and knowledge organized in quantitative models, such as forecasting models, simulation models, statistical models, Bayesian networks, and neural layers.
IV	The DMSS accesses highly semistructured data, information, and knowledge organized in knowledge chunks. Examples are if-then rules, if-then fuzzy rules, semantic networks, frames, scripts, and cases.
V	The DMSS accesses a network of highly ill-structured data, information and knowledge organized in knowledge bases. Examples are ontology–based repositories and distributed knowledge bases.

The processing axis describes the degree of embedded intelligence in the examined DMSS, as described in Table 1.5.

Table 1.5. Levels of processing (P) capability

Levels of P Dimension	Description of Capability
I	The DMSS provides all SQL-like actions: searching, adding, updating, deleting and sorting using a crisp logic mechanism. Also it supports all OLAP-alike actions such as drilling-down rolling-up, slicing and pivoting operations.
II	The DMSS provides all OLAP-alike actions: drilling-down, rolling-up, slicing and pivoting operations and/or all SQL-like actions of searching, adding, updating, deleting and sorting for fuzzy data.
III	DMSS provides services of classification, association, clustering, trend analysis and forecasting for quantitative data. Examples are neural networks, genetic algorithms, data-mining and statistical-based algorithms.
IV	DMSS provides services of algorithms and heuristics for complex analysis tasks with both qualitative and quantitative data, such as classification, diagnosis, interpretation and monitoring/control. Examples are rule-based inference algorithms, case-based techniques, and frame and semantic networks inference algorithms.
V	The DMSS provides services of algorithms and heuristics for complex synthesis tasks with both qualitative and quantitative data, such as discovering, explanation, planning, design and learning. Examples are agent-based behavioral algorithms and hybrid or integrated intelligent algorithms.

These tables, represent ordinal conceptual scales to measure the degree of: (i) user-interface capability, (ii) rawness in the data component and (ii) the degree of intelligence embedded in the algorithms or processing mechanisms of a particular DMSS. It must be noted that any support level includes or can include capabilities from previous levels. Hence, we claim that this framework is useful to analyze the support capabilities of past, current, and future DMSS such as i-DMSS.

A theoretical architecture of an i-DMSS - Figure 1.1 (Forgionne *et al.* 2002)- illustrates an example of how these capabilities could be implemented. Computational power and memory are needed to access and analyze massive data, and AI techniques are needed to develop the intelligent behaviors needed for complex decision-making situations.

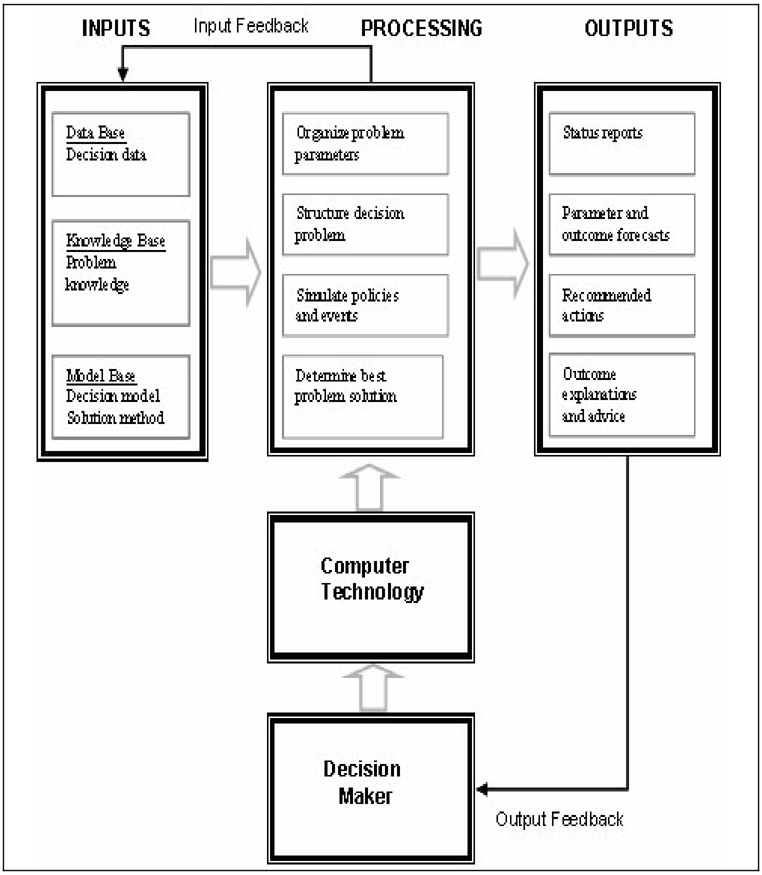


Figure 1.1. Conceptual i-DMSS architecture (Forgionne *et al.* 2002)

Methodology aside, then, intelligent systems have characteristics that are generally associated with human intelligence. Accordingly, we take the position that an intelligent system should demonstrate human intelligence as well as the underlying intelligence-generating mechanism. The more intelligent behaviors the system exhibits (*i. e.* the functionality), the more intelligent it is, but the behaviors must be caused by the utilization of underlying intelligent mechanisms (*i. e.* the structure).

1.3 Multicriteria Evaluation of i-DMSS

This chapter is focused on the evaluation of i-DMSS. Since these information systems are designed to support both the process of, and outcome from, decision-making, we propose a multi-criteria evaluation methodology. One such methodology is the Analytic Hierarchy Process (AHP).

1.3.1 Evaluation of i-DMSS in the Literature

A review of the literature and the criteria used for evaluation of i-DMSS is shown in Table 1.6.

Table 1.6. Criteria used for evaluation of i-DMSS in the literature

Authors	**Description of i-DMSS**	**No.**	**Evaluation Criteria**
Tsumoto, 2003	Internet-based medical DSS in telemedicine	1 2	- Enables doctors to take quick action - Agrees with experts classification accuracy
Wong, 2003	Proposes a pattern discovery approach in DSS to select a subset of data from large data sets	3 4 5	- Deeper understanding of the problem - More efficient planning - More efficient process controls
Palaniappan *et al.* 2002	DSS for an industrial process involving acrylic acid production process	6 7 8	- Correctly identifies hazards and wastes in process in agreement with expert practice - Helps decision maker assign realistic weights and preferences - Helps decision maker focus on critical areas of process to improve safety and environmental performance
Potter *et al.* 2002	DSS to assist in planning the schedule for spraying pesticides aerially	9 10 11 12	- Spray productivity - Spray efficiency - Fast solutions considering only near term events - Comparison to expert scheduler
Smith *et al.* 2001	Criteria for evaluation of DSSs for medicine	13 14 15	- Objective performance against a 'gold standard' such as actual survival records in terms of accuracy, precision and assessment of errors - Agreed measurements with experts - Subjective performance

Strachan *et al.* 2001	i-DMSS for a protection scheme for transmission network	16 17 18	-Optimize core design process time -Provide engineers with more time for checking and modification of designs -Fast and accurate final designs
Zeleznikow and Nolan, 2001	i-DMSS for property distribution following divorce in Australia, and to assist teachers in New York state to grade essays	19 20 21	-Agreement with domain expert -Save time in reaching a decision -Shared understanding of the problem
Cassaigne and Singh, 2001	Intelligent tactical DSS to enable firms to make superior pricing decisions in a competitive, dynamic environment	22 23 24 25	-Increase in sales units -Increase in cash turnover -Profit increase -Effectiveness of the process especially in terms of time
Nemati and Iyer, 1999	Asset allocation decision at a particular risk level	26	-Accuracy of predicting annual returns
Guerlain *et al.* 2000	Characteristics common to successful i-DSSs in practice by comparing 3 DSSs	27 28	-Accuracy of strategic prediction such as impact from building a new plant -Accuracy of tactical prediction such as scheduling or inventory management
Singh and Reif, 1999	Aids for electronic commerce	29	-Understanding of the implications of the alternatives
Chan *et al*, 2000	Flexible manufacturing system design aid	30	-Suitable design based on multi-criteria decision-making
Faye *et al.* 1998	i-DSS for short term water resource management of an irrigation system	31	-Outcome in terms of optimization in the presence of failed devices
Ifeachor *et al.* 1998	i-DSS for electronic fetal monitoring during labor and immediately after birth (DSS in routine use)	32 33	-Extensive comparison to expert judgment by experienced clinician -Economic benefit to hospital
Yang and Huang, 1996	i-DSS for transformer fault diagnosis	34 35	-Success in classification rates of transformer faulty conditions -Training time to establish the networks
Chan and Naghdy, 1997	i-DSS to assist the anaesthetist in body fluid balancing of a patient in surgery	36	-Comparison to expert opinion by examining records from surgery

Fazlollahi *et al.* 1997	Dynamic adaptation of decision support to select forecasting model for data for asset allocation	37 38 39 40	- Decision quality - User learning and understanding of problem - User satisfaction with process of decision making - Ability to generalize
Borenstein, 1998	i-DSS for design and evaluation of flexible manufacturing systems	41 42	- Improved process of decision making with systematic approach - Improved understanding of the problem
Lin *et al.* 1996	i-DSS to assist in investment in real estate	43 44	- Speed in decision making - Understanding of the problem
Kwok *et al.* 1996	Assist in assigning teaching duties in a secondary school	45 46 47	- Comparison to expert (school principal) - Save time - Reduce tedium in task
Renton and Wallace, 1996	Scheduling of cascade water management in hydro-electric plants	48	- Decision outcome of best generation profile based on optimal scheduling
Vraneš *et al.* 1996	i-DSS for capital investment decision	49	- Understanding of the problem
Pflughoeft *et al.* 1996	i-DSS for flexible manufacturing system scheduling	50 51 52 53	- Efficiency measured as speed of the decision - Performance improvement over current policy based on optimal solution - Ease of use - Portability
Kobbacy *et al.* 1995	Assist in optimization of maintenance routines of large technical systems	54	- Comparison to expert advice
Silverman, 1992	Provide expert critiquing system to criticize a user's proposed solution to a problem	55 56	- Improvement in task performance - Comparison to expert opinion

Similar criteria can be grouped into the process and outcome measures shown in Table 1.7.

Table 1.7. Process and outcome measures from the literature for i-DMSS

Process Measures	**Outcome Measures**
Supports real-time decision 1, 32	Improved organizational outcome 9, 10, 18, 22, 23, 24, 30, 31, 33, 34, 37, 48, 51
Enhanced understanding of the problem 7, 8, 21, 29, 38, 42, 44, 49	Comparison to a 'gold standard' 13

Ability to generalize 40	Comparison to expert opinion 2, 6, 12, 14, 19, 32, 36, 45, 54, 56
Faster decision 1,11, 16, 17, 20, 25, 35, 43, 46, 50	More accurate 26, 27, 28
More efficient 4, 5, 55	
Systematic approach 41	
User satisfaction 15, 39, 52	
Organizational satisfaction 47, 53	

1.3.2 A New Multicriteria Model for Evaluation of i-DMSS

The evaluation criteria in Table 1.7 can be further refined as shown in Tables 1.8 and 1.9. The process measures are shown in Table 1.9. Several of the criteria relate to Simon's classic model of the phases of decision-making discussed previously: *intelligence*, *design* and *choice*. Decision support systems may support one or more of the phases as the decision maker uses the system. In the intelligence phase, the user gathers information about the problem. Real-time systems, such as those offered by Tsumoto (2003), and Ifeachor *et al.* (1998), are designed to support the intelligence phase. Although Tsumoto and others did not explicitly include this criterion, we propose that it should be considered as part of the overall evaluation. In the design phase, the user develops the criteria and models important to the decision. Under this phase we include "enhanced understanding of the problem" as the user performs actions such as what-if scenarios. Choice is the phase in which the user is able to make a decision, and the "ability to generalize" is part of this phase.

The second process measure is efficiency. In the literature, efficiency is expressed in terms of time, cost and procedure. Procedural efficiency, as used here, expresses the concept that a systematic approach to decision-making can be codified in an i-DMSS. The thought processes associated with problem solving may be more efficient when considering decision problems that are similar to that in the i-DMSS. The final process measure is satisfaction. This criterion measures both the user's and organization's satisfaction with the i-DMSS. Such subcriteria as 'perceived ease of use' and 'perceived usefulness' from the technology acceptance model could be included under this area.

The criteria in Table 1.8 were used to develop two outcome measures as shown in Table 1.9. Organizational performance includes criteria such as increased profit and decreased cost that are often the stated reason for the i-DMSS. The second measure, predictive quality, measures the ability of the i-DMSS to match an authority in the decision environment. Two authorities are presented in the literature. The first is a 'gold standard', and this would include validated data that have been collected in the decision environment. An example might be robust data associated with survival rates under certain medical conditions. These data serve a similar function to physical data used to evaluate a physics-based model. The second type of authority is a domain expert. In complex decision tasks, experts are often used as the 'gold standard'. Examples are a physician in a medical i-DMSS or a lawyer in a legal i-DMSS.

Table 1.8. Process Measures to evaluate i-DMSS

Phase of decision making	**Criterion in evaluation of i-DMSS**
- Intelligence	Support real-time decision
- Design	Enhanced understanding of the problem
- Choice	Ability to generalize
Efficiency	
- Time	Faster decision
- Cost	More efficient
- Procedure	Systematic approach
Satisfaction	
- User	User satisfaction
- Organization	Organizational satisfaction

Table 1.9. Outcome measures to evaluate i-DMSS

Outcome	**Criterion in evaluation of i-DMSS**
Organizational performance	Improved organizational outcome
Predictive Quality	
- Gold standard	Comparison to a 'gold standard'
- Expert opinion	Comparison to expert opinion

Tables 1.8 and 1.9 are combined to form a multicriteria model for i-DMSS evaluation as shown in Figure 1.2.

In the evaluation, both the process of, and the outcome from, decision-making are considered. Not all criteria need to be applied to every model, and it is possible to add further subcriteria. The proposed model includes all of the criteria identified in the literature search. In Figure 1.3, the underlying AI mechanisms are represented by the UI, DIK, and P dimensions presented in Tables 1.3, 1.4 and 1.5. These dimensions affect the sub-criteria that determine proficiency, efficiency, and satisfaction measures that influence the process of decision making. In addition, the dimensions affect the organizational performance, "gold standard", and expert opinion measures that impact decision outcome.

Hence, the various measures of i-DMSS evaluation can be attributed to the underlying AI mechanisms created to support the decision-making process and outcome. Each dimension may have a differential impact on each measure. The P dimension, for example, may mainly impact the efficiency measures, while the DIK dimension may have a primary effect on proficiency. Nevertheless, each dimension is likely to affect all process and outcome measures to some degree. These underlying AI mechanisms are shown as an additional layer in Figure 1.3. In any given application, the actual i-DMSS will vary. Consequently, each dimension (UI, DIK, and P) can have varying capability levels (I, II, and so on, using the framework presented in Tables 1.3, 1.4, and 1.5).

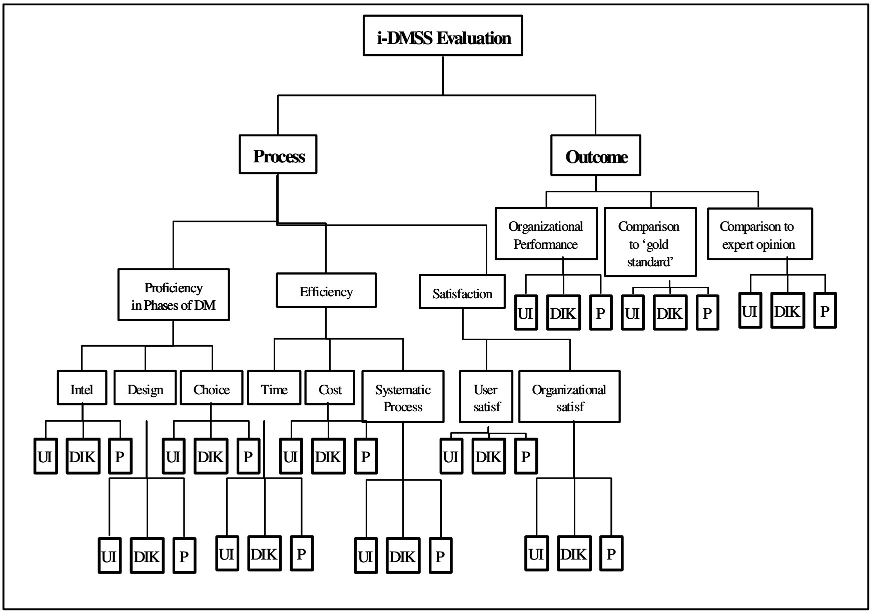


Figure 1.2. Proposed multicriteria evaluation model for i-DMSS Evaluation

By comparing the i-DMSS to a competing system(s) across the dimensions, we can assess the impact of the tested systems on the multiple criteria and thereby the process of, and outcome from, decision making.

1.4 Results and Discussion

The multicriteria model for i-DMSS evaluation can be quantitatively implemented using the analytic hierarchy process (Saaty 1977). The AHP has been shown to provide a comprehensive and unifying theory for the evaluation of decision-oriented information systems (Forgionne 1999, Forgionne and Kohli 2001). The AHP provides a logical and scientific basis to decision making in which pairwise comparisons of components are made with respect to a common goal or criteria (Saaty 1977, Harker 1988). The decision problem is structured as a hierarchy of criteria, subcriteria and alternatives, with the number of levels being determined by the problem. Once the hierarchy is established, the alternatives are evaluated by pairs with respect to the criteria on the next level. The criteria can be weighted, if desired, according to the priority of each criterion.

As an illustration, we consider the i-DMSS by Palaniappan *et al.* (2002) for an industrial process involving acrylic acid production process including integrated safety and waste minimization analysis during process design. An expanded version of the evaluation measures shown in Table 1.8 and taken directly from the research were that the i-DMSS:

(1) Results in substantial reduction in time and effort by obviating the need for detailed analysis in some cases, *i. e.* synergistic alternatives, while highlighting such a need in others, *i. e.* tradeoffs (p. 768);
(2) Provides a design approach that is inherently safer, environmentally friendlier, simpler and cost effective (p. 757);
(3) Provides enhanced decision making (p. 760);
(4) Enables the designer to understand the influence of change in design features on safety and environmental performance (p. 768);
(5) Allows improved documentation (p. 760);
(6) Increases the process understanding of decision maker through identification of issues and synergies and tradeoffs among alternatives (p. 760);
(7) Implicitly incorporates the iterative nature of process design, providing a systematic and unified framework (p. 767);
(8) Provides methods for ranking and prioritization of alternatives based on experience and judgment of concerned parties (p. 767);
(9) Correctly identifies hazards and wastes in process in agreement with 'gold standards' such as P-graph analysis (p. 773);
(10) Identifies correlations between alternatives and classifies them, enabling the designer to prioritize them (p. 773);
(11) Helps decision maker assign realistic weights and preferences (p. 773);
(12) Helps decision maker focus on critical areas of process to improve safety and environmental performance in later stages of the design (p. 773);
(13) Integrates P-graph, digraph and functional models to represent the cause-and-effect relationship of materials, reactions and separations involved in the process and interactions between process variables in different units (p. 773);
(14) Chemical process industries can bring products to market at low lifecycle costs without compromising on safety and environmental standards (p. 757);
(15) Decision makers increase their process understanding and learn appropriate weight assignments (p. 767);
(16) Methodology implicitly incorporates the iterative nature of process design (p. 767).

These measures can be identified with the multicriteria model for i-DMSS evaluation shown in Figure 1.3. The results are shown in Table 1.10. The i-DMSS supports both the process of, and the outcome from, decision making. As with many i-DMSS, the process of decision-making is significantly affected and is as important as the outcome measures often used to evaluate traditional DMSS.

The capability assessment framework provides a description of the three dimensions for the Palaniappan *et al.*'s i-DMSS: (i) user interface (UI); (ii) data, information and knowledge (DI&K); and (iii) processing (P). The user interface uses structured commands and menus. The presentation language is based on texts and animated graphics such as digraph models. "A graphical user interface enables the user to input process-specific information, view and edit the models and browse the … analysis results" (p. 768). Using Table 1.3, the UI is classified as I. The i-DMSS is described as having an object-oriented architecture that is implemented as an expert system (p. 768). Both process-specific and process-general knowledge are represented. "Domain heuristics are derived from chemical engineering principles, inherent safety index, inherent safety and waste minimizations principles, whereas, digraph and functional models are based on chemical engineering principles" (p 768). The information is semi-structured, and the inference engine systematically processes the knowledge. According to Table 1.4, the DI&K representation has level IV. In terms of processing, the i-DMSS uses "heuristic rules that can be applied to process units to derive top-level alternatives" (p. 764). This is supplemented by analysis for a cause and effect representation of each material in the process. Alternatives are generated and decision-making tools such as cost-benefit analysis are used to prioritize the alternatives. Values and preferences are assigned by the user based on experience and judgment. The P level can be classified as IV in Table 1.5.

Table 1.10. Evaluation Measures for the Palaniappan, Srinivasan and Halim (2002) i-DMSS

i-DMSS Evaluation Measures	Palaniappan *et al.*'s evaluation measures
Process of decision-making	
* *Proficiency in Phases*	
- Intelligence	13
- Design	4, 6, 11, 12
- Choice	10
* *Efficiency*	
- Time	1
- Cost	2
- Procedure	7
* *Satisfaction*	
- User	3, 15
- Organization	5, 16
Outcome of decision making	
* *Organizational performance*	14
* *Gold standard*	9
* *Expert opinion*	8

The evaluation model can be implemented quantitatively using the AHP model delivered through software such as Expert Choice (2004). In order to provide numeric values for the model, a comparison between two alternatives is needed, for example, a comparison between an i-DMSS and a normal DMSS. In Expert Choice, each dimension, subcriteria, criteria, and measure can be evaluated, in turn, with respect to the two alternatives by indicating how much more one alternative is preferred with respect to the subcriteria than the other. The method then will transform the comparisons to eigenvalue scales that are consolidated into decision values for the alternative systems.

The specific research with the i-DMSS by Palaniappan *et al.*'s (2002) does not provide sufficient data to actually evaluate the AHP model in this chapter. However, if data were available, it would be possible to determine the priorities associated with the two alternatives. The AHP model will then generate decision values for the alternatives, *i. e.* i-DMSS and No_i-DMSS, which will indicate if the i-DMSS has value compared to its alternative. In addition, it is possible to determine which artifacts led to the decision value, and which AI techniques contributed to the value of the i-DMSS. Further, using a stochastic enhancement of AHP, it is also possible to statistically identify the criterion (or criteria) that most significantly contributes to the benefit (or lack thereof) of the i-DMSS as well as determining if the i-DMSS offers a statistically significant improvement over the alternative (Phillips-Wren *et al.* 2004). These methods provide powerful analysis techniques and insight into the role of AI in providing decision value in i-DMSS.

1.5 Conclusions and Contributions to the Literature

In this chapter we explored the concept of human and artificial intelligence as they relate to aiding decision making. Then, a set of specific categories were identified that were organized and integrated through an architecture for evaluating i-DMSS, by acknowledging the specific characteristics provided by intelligent systems rather than decision support systems in general. Next, using the new model posited, it was demonstrated that an empirical evaluation of an i-DMSS can be accomplished. Furthermore, it was suggested that, given sufficient data, the analytic hierarchy process can be used to quantitatively evaluate a system.

This chapter also suggested that although outcome measures have traditionally been the figure of merit for traditional DSS, i-DMSSs have had much greater impact than has been recognized by the community. Since i-DMSSs implement intelligent characteristics, they affect both the process of, and outcome from, decision making. However, any investment effort completed by the addition of these AI-based components in i-DMSSs should be linked to measurable improvements besides simply computational speed and gains in data quantity. Hence, selecting the best architectural design of several i-DMSS candidates based on ranks assigned by their contributions to the decision-making process, as well as to outcome, this chapter contributes to the literature through: (1) the recognition of the impact of AI on i-DMSS; (2) the identification of the criterion (or criteria) used to evaluate i-DMSS in the literature; (3) the categorization of the evaluation measures; (4) an architecture for evaluation of i-DMSS; and (5) the recommendation of a multicriteria model to assess i-DMSS.

Interesting research issues such as: (a) the inclusion of predictive components so that the model can be used as a priori evaluation method instead of a post-mortem evaluation; (b) a refinement of the abstract multi-layer architecture from AI-based techniques and the tasks included in the decision-making process, and (c) a refinement of the relationships between the AI-based component and decision-making task layers, are suggested finally for further investigation.

Acknowledgements

The authors would like to thank the anonymous reviewers whose comments and recommendations significantly contributed to the final manuscript.

References

Adelman L (1992) Evaluating Decision Support and Expert Systems, John Wiley & Sons, Inc., New York, NY.

Albus J, Meystel A (2001) Engineering of Mind, Wiley Series on Intelligent Systems, John Wiley and Sons, Inc., New York, NY.

Boering E (1933) The Physical Dimensions of Consciousness. New York: Century.

Borenstein D (1998) IDSSFLEX: an intelligent DSS for the design and evaluation of flexible manufacturing systems. Journal of the Operational Research Society 49(7): 734-744.

Brooks F (1996). The computer scientist as toolsmith II. Communications of the ACM, 19(3): 61-68.

Brown C, O'Leary D (1995) Introduction to Artificial Intelligence and expert systems. Accessed from http://accounting.rutgers.edu/raw/aies/www.bus.orst.edu/faculty/brownc/ es_tutor/ es_tutor.htm, on September 15, 2004.

Cassaigne N, Singh MG (2001) Intelligent decision support for the pricing of products and services in competitive consumer markets. IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews) 31(1): 96-106.

Chan FTS, Jiang B, Tang NKH (2000) The development of intelligent decision support tools to aid the design of flexible manufacturing systems. International Journal of Production Economics 65(1): 73-84.

Chan W, Naghd F (1997) An intelligent decision support system for body fluid balancing. In: IEEE International Conference on Intelligent Processing Systems, New York, NY, pp 1537-1541.

Dhar V, Stein R (1998) Intelligent Decision Support Methods, Prentice-Hall, Upper Saddle River , pp 7-14.

Elam J, Konsynski B (1987) Using artificial intelligence techniques to enhance the capabilities of model management systems. Decision Sciences, 18: 487-501.

Eom S, Lee S, Kim E, Somarajan C (1998) A survey of decision support systems applications (1998-1994). Journal of Operational Research Society, 49: 109-120.

Eom S (1998) An overview of the contributions to the decision support systems area from artificial intelligence, Proceedings of the ICIS 1998 Conference, 149-151.

Expert Choice (2004) Accessed from http://www.expertchoice.com/ on November 1.

Faye RM, Mora-Camino F, Sawadogo S, Niang (1998) An intelligent decision support system for irrigation system management. In: IEEE International Conference on Systems, Man, and Cybernetics; New York, NY, pp 3908-3913.

Fazlollahi B, Parikh MA (1997) Adaptive decision support systems. Decision Support Systems 20(4): 297-315.

Forgionne G (1999) An AHP model of DSS effectiveness. European Journal of Information Systems 8: 95-106.

Forgionne G, Kohli R (2001) A multiple criteria assessment of decision technology system journal qualities. Information Management 38: 421-435.

Forgionne G, Mora M, Cervantes F, Gelman O (2002) I-DMSS: A conceptual architecture for the next generation of decision-making support systems in the internet age. In: Adam F, Brézillon P, Humphreys P, Pomerol J (eds), Decision-making and Decision Support in the Internet Age, Proceedings of the DSIage2002, IFIP WG 8.3, Cork, Ireland, July 4-7, pp 154-165.

Goul M, Henderson J, Tonge F (1992) The emergence of artificial intelligence as a reference discipline for decision support systems research, Decision Sciences, 23, pp. 1263-1276.

Gottinger HW, Weimann P (1992) Intelligent decision support systems. Decision Support Systems, 8(4): 317-332.

Grabowski M, Sanborn S (2001) Evaluation of embedded intelligent real-time systems. Decision Sciences (Winter) 32(1): 95-123.

Gray P, Watson H (1996) The new DSS: data warehouses, OLAP, MDD and KDD. In: Proceedings of the AMCIS Conference 1996, Phoenix, AZ, USA .

Guerlain S, Brown DE, Mastrangelo C (2000) Intelligent decision support systems. In: IEEE International Conference on Systems, Man and Cybernetics.`Cybernetics Evolving to Systems, Humans, Organizations, and their Complex Interactions', Piscataway, NJ, pp 1934-1938.

Harker P (1988) The Art and Science of Decision-making: The Analytic Hierarchy Process. Working Paper 88-06-03, Decision Science Department, The Wharton School, University of Pennsylvania, Philadelphia, PA.

Holsapple CW, Whinston AB (1996) Decision Support Systems, West Publishing Company, St. Paul, MN.

Ifeachor EC, Garibaldi, JM, Skinner J (1998) Intelligent decision support tools for the management of labour. In: IEE Colloquium on Intelligent Decision Support in Clinical Practice, pp 1-7.

Jensen A (1998) Does IQ matter ? Comentary, pp. 20-21, November. Comments to F. Chabris, "IQ since the Bell Curve", Commentary, August.

Jensen A (1999) The G Factor: the Science of Mental Ability. Psycoloquy, 10, #23.

Jensen A (2000) Artificial Intelligence and G Theory concern different Phenomena, Psycoloquy, 11, #86.

Kobbacy KAH, Proudlove NC, Harper MA (1995) Towards an intelligent maintenance optimization system. Journal of the Operational Research Society 46(7): 831-853.

Kwok LF, Lau WT, Kong SC (1996) An intelligent decision support system for teaching duty assignments. In: Narasimhan VL, Jain LC (eds), Proceedings of the Australian New Zealand Conference on Intelligent Information Systems, pp 97-100.

Lin R, Wang Q, Hu J (1996) An intelligent decision support system applied to the investment of real estate. In: Proceedings of the IEEE International Conference on Industrial Technology, New York, NY, 801-805.

Macintosh A (2004) The optimization of What ?. Proceedings of the DSS2004 Conference, Prato, Italy. Keynote Address.

Maynard S, Burstein F, Arnott, D (2001) A multi-faceted decision support system evaluation approach. Journal of Decision Systems, special issue "DSS in the New Millennium", 10(3-4): 395-428.

McCarthy J (2003) What is artificial intelligence ?. Technical Report. Computer Science Department, Stanford University. Accessed from http://www-formal.standord.edu/jmc on September 9.

Mora M, Forgionne G, Gupta J, Cervantes F, Gelman O (2003) A framework to assess intelligent decision-making support systems. In: Proceedings of the 7th International Conference KES 2003, Oxford, UK, September (Lecture Notes on Artificial Intelligence 2774, Springer-Verlag), 59-65.

Nemati HR, Iyer LS (1999) An intelligent decision support system prototype for asset allocation. In: Haseman WD, Nazareth DL (eds), Proceedings of the Fifth Americas Conference on Information Systems; Atlanta, GA, 70-72.

Newell A, Simon H (1976) Computer science as empirical inquiry: Symbols and Search (1975 ACM Turing Award Lecture), Communications of the ACM, 19(3): 113-126

Palaniappan C, Srinivasan R, Halim I (2002) A material-centric methodology for developing inherently safer environmentally benign processes. Computers & Chemical Engineering 26(4-5): 757-774.

Parnas D (1985) Software aspects of strategic defense systems, ACM SIGSOFT Software Engineering Notes, 10(5): 15-23.

Pflughoeft KA, Hutchinson GK, Nazareth DL (1996) Intelligent decision support for flexible manufacturing: Design and implementation of a knowledge-based simulator. Omega 24(3): 347-360.

Phillips-Wren G, Forgionne G (2002) Evaluating web-based and real-time decision support systems. In: Adam F, Brézillon P, Humphreys P, Pomerol J (eds), Decision-making and Decision Support in the Internet Age, Proceedings of the DSIage2002, IFIP WG 8.3, Cork, Ireland, July 4-7, 166-175.

Phillips-Wren G, Hahn E, Forgionne G (2004) A multiple criteria framework for the evaluation of decision support systems. Omega 32(4): 323-332.

Pohl J (2005) Intelligent software systems in historical context. In: Phillips-Wren G, Jain LC (eds), Intelligent Decision Support Systems in Agent-Mediated Environments, IOS Press: The Netherlands, 3-34.

Potter W, Ramyaa LJ, Ghent TT (2002) STP: an aerial spray treatment planning system. In: Proceedings of the IEEE Southeast Conference, pp 300-305.

Proudlove NC, Vedera S, Kobbacy KAH (1998), Intelligent management systems in operations: a review. Journal of the Operations Research Society, 49: 682-699.

Renton MW, Wallace AR (1996) Expert system scheduling of cascade hydro-electric plants. Proceedings of the International Conference on Opportunities and Advances in International Power Generation, IEE, London, UK, pp. 69-72.

Saaty TL (1977) A scaling method for priorities in hierarchical structures. Journal of Mathematical Psychology 15: 234-281.

Silverman BG (1992) Evaluating and refining expert critiquing systems: A methodology. Decision Sciences 23(1): 86-110.

Simon H (1996) Models of My Life. MIT Press.

Simon H (1997) Administrative Behavior, Fourth edition (Original publication date 1945), The Free Press, New York, NY.

Singh R, Reif HL (1999) Intelligent decision aids for electronic commerce. In: Haseman WD, Nazareth DL (eds), Proceedings of the Fifth Americas Conference on Information Systems (AMCIS), Atlanta, GA, 85-87.

Smith A, Nugent C, McClean S (2001) Intelligent decision support systems for medicine: inherent performance evaluation. In: Conference Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Piscataway, NJ, vol. 4, 3746-3749.

Sprague RH (1980) A framework for the development of decision support systems. MIS Quarterly, 4(4): 1-26.

Strachan SM, West GM, McDonald JR (2001) Knowledge management and intelligent decision support for protection scheme design and application in electrical power systems. In: Seventh International Conference on Developments in Power System Protection, London, UK, 559-562.

Tsumoto S (2003) Web based medical decision support system: application of Internet to telemedicine. In: Proceedings of the Symposium on Applications and the Internet Workshops, Los Alamitos, CA, 288-93.

Turban E, Aronson J (1998) Decision Support Systems and Intelligent Systems. A. Simon and Schuster Company, Upper Saddle River, NJ,

Turing A (1950) Computing machinery and intelligence. Mind 59, 433-460.

Vraneš S, Stanojević M, Stevanović VL (1996) INVEX: investment advisory expert system. Expert Systems 13(2): 105-120.

Wong AKC, Yang W (2003) Pattern discovery: a data driven approach to decision support. IEEE Transactions on Systems, Man and Cybernetics, Part C 33(1): 114-124.

Yang H, Huang Y (1996) Intelligent decision support for diagnosis of incipient transformer faults using self-organizing polynomial networks. In: Proceedings of the 20th International Conference on Power Industry Computer Applications, New York, NY, pp 60-66.

Zeleznikow J, Nolan JR (2001) Using soft computing to build real world intelligent decision support systems in uncertain domains. Decision Support Systems 31(2): 263-285.

2

On the Legacy of Herbert Simon and his Contribution to Decision-making Support Systems and Artificial Intelligence

Jean-Charles Pomerol[1] and Frédéric Adam[2]

[1] Université Pierre et Marie Curie, 4 Place Jussieu, 75252 PARIS 5, France, e-mail: Jean-Charles.Pomerol@lip6.fr
[2] Business Information Systems, University College Cork, Cork, Ireland, e-mail: fadam@afis.ucc.ie

Herbert Simon is a key researcher in a number of areas of the IS field. Although his work has been used as like foundation for much research in DMSS, it is in the area of artificial intelligence that he has most directly contributed to the progress of scientific thinking. In this chapter, we examine his legacy and illustrate the impact of his research on DMSS and AI.

2.1 Introduction

Since the late 1940's, Herbert Simon has been unequivocally associated with management, decision-making and artificial intelligence, which he co-founded with A. Newell. None of his contemporaries have had such a far-reaching impact on management, especially when his further work with James March is considered. Mintzberg himself, who considerably advanced research on management practice, stated that he always considered Simon to be the most influential and important contemporary author in terms of organizational theory (1990, p. 94). In terms of artificial intelligence, Simon was the first to propose the notion of an intelligent problem-solving device and, in collaboration with Newell, he provided a blueprint for how such a device could be developed and how it would operate.

This legacy leads us to review Herbert Simon's contribution to the decision-making, decision support and Artificial Intelligence areas and to show how the science and practice of managerial decision-making and the development of intelligent systems have changed under his influence. We also consider to what extent his work, notably his pioneering research into the decision-making process within economic organizations (for which he received the Nobel Prize in 1978), contributed to the establishment of DMSS as a field of research. This chapter first considers the new ideas brought by Simon in management theory and then looks at his contribution to our understanding of managerial decision making and DMSS.

Finally, it reviews Simon's AI project and what followed from it. The chapter concludes by raising some questions for designers of intelligent DMSS.

2.2 The Manager as a Decision Maker

From the observations gathered during his years working for the city government in the town of Milwaukee and from his teaching and lecturing at the University of Chicago, Simon conceptualized the need for a science of management, a science that should ideally be as falsifiable as the other sciences. Recognizing that he found himself at the very beginning of this ambitious project (Simon 1997, p. xi) , Simon saw himself as one of the pioneers of the second generation of scientific management, after Fayol, Taylor and others. It was above all Taylor who attracted his attention. Taylor (1911) published "The Principles of Scientific Management", a book dealing mainly with the improvement and effectiveness of production processes and the role of human labor in the elementary operations in production (Simon 1997, p. ix). It was doubtless no coincidence that Simon's book on the introduction of computer science into management was entitled "the New Science of Management Decision". In the foreword of the 1977 edition, Simon actually wrote (ibid. p. x) "*The computer and the new decision-making techniques associated with it are bringing changes to white-collar, executive and professional work as momentous as those that the introduction of machinery has brought to manual jobs*".

Simon's basic idea, as expressed in "Administrative Behavior", is that the correct angle from which to approach a study of organization management is that of the decision and the action that follows (ibid. p. 1). Thus, the manager must primarily be viewed as a decision maker (Simon 1977, p. 39). This is well characterized in the book with March (1993, p.3): "*The central unifying construct of the present book is not hierarchy but decision-making, and the flow of information within organizations that instructs, informs, and supports decision-making processes*". This became the unifying thread in Simon's future work on decision-making and Simon described himself (Simon 1991, p. xvii) as somebody "*who has devoted his scientific career to understanding human choice*". Thus, Simon's ambitious program was to understand organizations and their management as an aggregate of human choices; not like in economics theory based on the abstracted behavior of homo œconomicus, but based on the real behavior of people, that is to say considering how those involved in making decisions acquire the necessary information, how they perform their calculations, or more importantly still whether they are capable of correctly evaluating the consequences of the decisions, according to the events, as postulated by the maximization of utility (Simon 1997, p. 20). In parallel, he sought to understand human decision-making to the extent where it would become possible to model and reproduce such decision making using automatons.

A key consequence of Simon's observations and ideas is that decisions and the actions that follow them cannot easily be distinguished. Thus, decision-making support systems (DMSS) should primarily be geared as models for action, but action in an organization is a cascade at intertwined subactions and consequently DMSS design must accommodate human reasoning in a variety of levels, from the strategic level to the lowest level of granularity of action decided by managers. However, we believe that this has not been applied in the practice of DMSS development, and that DMSS have focused on high-level decision making (strategic decision) but using low levels of representation (data, equation, etc) because (1) the notion of representation level has not been sufficiently studied and (2) high level decisions are more appealing than small decisions (Humphreys and Berkeley 1985, Pomerol and Adam 2003b).

2.3 Decision Process and Intelligent Decision Support

To scientifically deal with his analysis of the decision process, Simon began by distinguishing between facts and values (Simon 1997, ch. 3) or what is and what ought to be. Facts are what can be verified or falsified, whereas values are the objectives of the decision maker and, beyond this, his actual wishes. It follows that we can only evaluate a decision if we know the objectives of the decision maker (ibid. p.56 et seq.) This notion, reminiscent of the idea of aspiration level introduced by Dembo (see Lewin *et al.* 1944), was adopted by Simon and became an important feature of his understanding of the "heuristic" search that managers go through when making decisions (see section 3.3). Many interactive methods in decision making rely on the notion of local adjustments according to the level of satisfaction reached at every given step. This is a basic tenet of "bounded rationality" (see Selten 2002). Thus, to evaluate the quality of a decision, researchers must know the utility of the decision maker, and understand what he or she expects in terms of the probabilities of future events (this aspect was not specifically studied by Simon).

2.3.1 Towards a Model of the Decision-making Process

Simon observed that the problems that trigger decisions are not factual data but constructs. In his own words: "*problems do not come to the administrators carefully wrapped in bundles with the value elements and the factual elements neatly sorted*". Secondly, he observed that decision "*is a matter of compromise*", *i. e.* all decision makers have several more or less contradictory objectives in mind. Thus, Simon was the first to stress the multicriteria aspect of managerial decisions.

Based on these observations, Simon (1997, p. 77) laid the foundations for his seminal model of decision making. He broke down decision making as follows:

1. identify all the possible alternatives;
2. determine all the possible consequences of these alternatives;
3. evaluate all the possible consequences.

In contrast with the simplistic view of Dewey, Simon is interested in the mechanics of the decision-making process, in that he considers how a decision maker evaluates all the consequences and compares them with each other. This is a central problem in any decision process in that evaluating consequences requires that managers have a complete knowledge of all future events and their probabilities.

Simon also attempted to build three key factors of cognitive load into his model of decision making: attention, information and stress. Given the limited cognitive capacity of humans, attention is a limited resource that plays an important part in decision making. This theme is central in Simon's work with March ("… *the ways in which attention is allocated are critical to understanding decision*" (1993, p. 4)). Cognitive limitations play a substantial role in the concept of bounded rationality in that, as Simon stressed, they preclude the exhaustive study of all of the alternatives and their consequences. This led Simon to present his famous four phases (Simon 1977): intelligence, design, choice and review.

Simon was aware of the interdependence between the phases and he provided examples of feedback from one stage into another. He also indicated that each stage can be considered recursively as a decision in itself (ibid. p. 43). Thus, his model enabled him to eliminate the common problem faced by decision-making researchers thus far, for instance, the artificial reduction of decisions to the moment of choice (see also Langley *et al.*, 1995). He said: "*All the images falsify decision by focusing on the final moment*" (Simon 1977, p. 40). This new idea was to bring the study of decision-making out of the mythology of management stories and link it firmly to the field of information systems.

The Simonian view of decision phases is well known and has been used in many studies of decision-making processes in organizations (Pomerol 1994). The advantage of such a break down of decisions is that it enables a progressive study of what is otherwise a very complex process. It also emphazises that the first two phases: intelligence and design, act as boundaries or even as constraints as in Simon (1977), on the response of organizations. Obviously, alternatives that have not been investigated are much less likely to go ahead than courses of action that have been well documented. In addition, good decisions are unlikely to be reached if the search for assumptions has not been carried out properly (Pomerol 1994). This is comprehensively illustrated in Janis' (1972) description of how the Bay of Pigs decision made by the Kennedy administration turned out to be one of the worst fiascos in recent history because (Janis 1972, p. 14):

> *"all the major assumptions supporting the plan were so completely wrong that the venture began to founder at the outset and failed at the early stages"*

Janis further commented that if the President and his advisors had imagined that the 'nightmarish' scenario that actually unfolded could materialize, or even, if they had simply considered that it could happen, they would surely have rejected the project outright (Janis, 1972).

2.3.2 Programmable and Non-programmable Decisions

As Simon's thoughts gradually turned toward the computer, he introduced another oft-quoted aspect of decision: the distinction between programmed and nonprogrammed decisions (Simon 1977). He stated: "*Decisions are programmed to the extent that they are repetitive and routine, to the extent that a definite procedure has been worked out for handling them so that they don't have to be treated from scratch each time they occur*" (Simon 1977, p. 46). Decisions on the other hand are nonprogrammed "*to the extent that they are novel, unstructured and unusually consequential*" (Simon 1977, p. 46). The fundamental unity of Simon's thinking is evident here, for organizations, like computers, are systems designed for "complex information processing" (Simon 1977, p. 15). The processing of information for decision is the key to the whole of Simon's work. Programmed decisions obey computer programs or other programs that are computerizable, while nonprogrammed decisions come under the heading of "problem solving" (Simon 1977, p. 64-65). Thus, programmed decisions can be modeled in DMSS whereas problems that cannot be modeled are outside the realm of DMSS. From these ideas stems the classical belief that semiprogrammed decision are the real target of DMSS and that DMSS must be interactive so the decision maker can complement the part of the model that is not "structurable".

The issue of determining whether a decision lends itself to programming is at the core of the concept of organizational learning, which then became widely investigated, in particular by March. The issue of recognizing 'decision patterns' also emerged and led to 'case-based reasoning', which became a recurrent research theme in artificial intelligence (Pomerol 2001).

2.3.3 Heuristic Search

From the very beginning the artificial intelligence (AI) project was intended to design "intelligent" systems, that is to say: systems mimicking human beings engaged in highly skilled tasks. In this sense, intelligent DMSSs are connected to AI, but they are rarely mentioned as such in mainstream AI literature. However, decision making, is a very specific human trait. As Damasio *et al.* (1996) stated:

> *"Decision-making is, in fact, as defining a human trait as language"*

Consequently, decision making is a fundamental component of "human problem solving". In the first studies on "Human Problem Solving", which are recorded in the seminal book of Newell and Simon (1972), decision making is intertwined with heuristic search. The first intelligent problem solver, namely GPS (see Newell and Simon 1963, Newell and Simon 1972), was created on the principle of recursive difference reduction. In other words, starting from a final goal the system reached a series of intermediary subgoals which were closer to the current state than the final goal, but which moved towards the final goal (figure 2.1).

The core principle of a heuristic search is to make the decision about the "best intermediary sub-goal". Once engaged on a path, there are actually only three possible decisions in a heuristic search (figure 2.2):

- develop, *i.e.* search new sub-goals along the same path for further reducing the difference without changing the current state;
- backtrack, *i.e.* change the current path by returning to an already explored state;
- continue, *i.e.* change the current state along the same path.

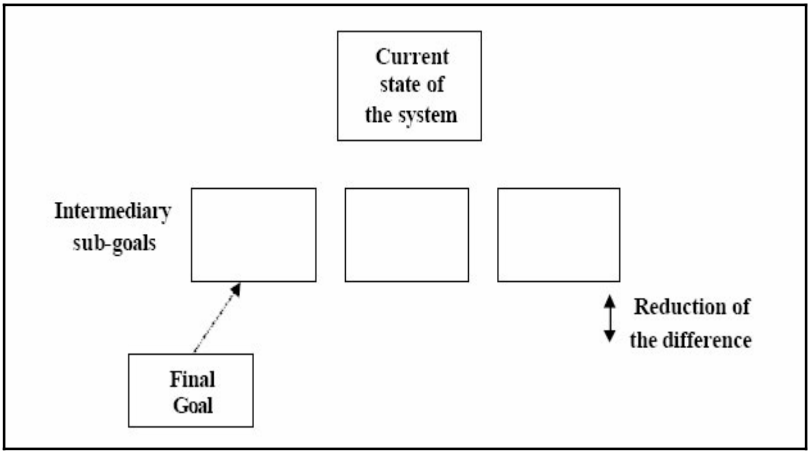


Figure 2.1. Intermediary sub-goals are closer to the current state than the final goal

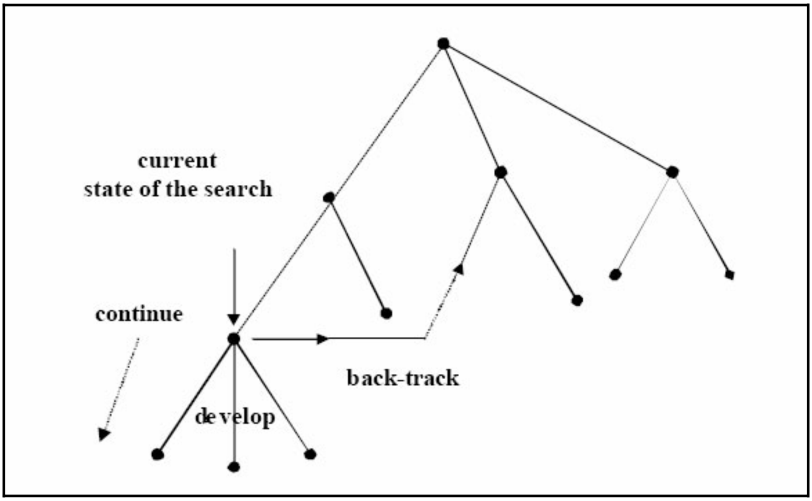


Figure 2.2. Heuristic decisions in tree format

Thus, decision making is at the heart of the heuristic search process. In GPS the decisions were made based on "difference tables" that heuristically assessed the differences between goals. These tables were filled by the designer of the system.

The focal point of any heuristic search is that the path is not defined *ex ante* but that a method exists to develop such a path according to the decisions made by the system during the process, or by the user in the case of an interactive search. In principle, these decisions rely exclusively on the evaluation of the difference between a current state of the system and a better state to be attained (goal). This reflects the fundamental orientation that decision and action are always motivated by the desire to reach a better state of the world. The use of the word "evaluation" is quite important: it means that any decision system – live or artificial – has one or several evaluation functions, that they use in order to determine their preferred options. In a heuristic search the evaluation function assesses the value of each node (each state of the world).

In the case of purely human decision making, the decision maker evaluates the current state according to the possible courses of actions (what-if analysis) and the consequences of all the possible decisions (Pomerol 2001). These consequences are, of course, assessed according to the preferences of the decision maker. This means that any artificial system must possess a model of the preferences of the end user. This is possible in industrial contexts, because the characteristics of each item to be produced or process to be monitored are known in advance and the evaluation of the outputs of the system according to these characteristics is relatively easy. The situation is dramatically different in the case of management where the preferences of the organization or of the decision maker are complex, diverse and inconsistent enough that are very difficult, if not impossible, to model. This is why in most decision support systems interactivity is used to give the opportunity to the decision maker to remain in the loop and control the decisions made by the system (Lévine and Pomerol 1989, 1995). In other words the model used by the system is incomplete because the evaluation (in fact the preferences) of the decision maker is not modeled.

Thus the first contribution of AI to DMSSs is to provide an understanding of the structure of DMSSs as information processing system (IPS), very similar to any "intelligent" system *à la* Newell and Simon (Bonczek *et al.* 1981). A further contribution is the emphasis placed on the fact that any intelligent system needs an evaluation function, and that the role of the decision maker in interactive systems, and especially DMSSs, as the person who directs the heuristic search by introducing his/her preferences is crucial to the outcome of the process (Lévine and Pomerol 1989, 1995). As an artificial system, a DMSS system is incompletely modeled.

2.3.4 Representation Levels

In tackling any new decision situation, there are several levels of representation of the problem at hand. At the top there is a general context: for example, the problem faced is a legal one, or one that involves logistics, human resources or any other area. This general level being identified, the choice of a problem-solving method can be made, for example operational research for a logistics problem. Then a model can be created, including the selection of the parameters involved in the model.

Complete descriptions of the concept of representation levels and how they apply to developing DMSSs can be found in Humphreys and Berkeley (1985), and also in Pomerol and Adam (2005), the purpose of this section is merely to emphasize how the model or models resulting from the process can be embodied in a computerized system. This is done at the two lower levels of representation, where designers code their models and input the data. The data are the values users change when they modify a value in a cell in a spreadsheet software (*e. g.* EXCEL). The model is changed and updated when users modify the formulae in the spreadsheet software. These are two different levels of representation.

The main virtue of DMSSs is that they are systems that allow an heuristic search (exploration) at two levels: models and data (Lévine and Pomerol 1995, Pomerol and Adam 2003b, Pomerol and Adam 2005). This led researchers who described DMSSs to propose the classical and compulsory architecture of a DMSS with a data base and a model base (Sprague and Carlson 1982). Thus DMSSs are at the crossroad of AI and cognitive science, in that they rely on a mixture of heuristic search performed at various representation levels. As quoted in the book of Newell and Simon, "*the secret of decision-making is that there is not secret*": very simple heuristics are at work both in human brain and artificial systems, but the representations they are based on are highly sophisticated and multi-level.

2.3.5 Decision = Recognition + Reasoning

Based on the material presented in the previous four sections, the reader could justifiably conclude that decision making is a mixture of: representations, heuristic searches, "what-if" analysis, scenario development (Pomerol 2001) and outcome evaluation. This would already constitute a broad understanding of what decision making is about, but, in AI terms, decision-making is even more complicated because the first step in the process we described is the recognition of external stimuli or diagnosis of the current state of the world. This recognition is also identification, classification and interpretation of the various signals that human decision makers can pick up in the environment. The system component that is designed for this recognition task is called "interpreter" in Newell and Simon's book. It has been an interesting and important observation that many intuitive decisions are triggered immediately after the recognition stage. Klein (1993) coined the term "recognition-primed decisions" to describe this phenomenon, which we can probably label decision by intuition. This type of decision has been modeled in AI under the name of case-based decision systems (Gilboa and Schmeidler, 1995, 2000a, 2000b).

When such a pattern of decision making is used, the recognition of a given pattern can automatically trigger a suitable decision. This is relatively straightforward in AI terms and Simon (1995a and 1995b) has provided us with some insights in how to develop such systems mimicking intuition and inspiration. This aspect of pattern recognition has been largely neglected in DMSS design and we are not aware of many case-based DMSSs (see some references in Pomerol 2001 and in Pomerol 2005) that have been experimented with.

2.3.6 Intelligent Decision Support

An intelligent DMSS, like any information processing system (however intelligent) is made-of an interpreter, a reasoning system for symbolic calculus (Newell and Simon, 1976), memories and action triggers. Thus, any attempt at designing a i-DMSS must consider these ingredients as components.

At the initial stage of the process of decision making, the first tasks that must be supported by DMSS are data acquisition and interpretation. This is a weakness of many i-DMSS because, in the case of many clerical tasks, acquisition is made by simultaneously reading and interpreting, and machines are not so good at reading especially with manuscript documents or with images that must be interpreted. At this level, pattern and case recognition are modeled by case-based reasoning and can produce simple decision systems in which patterns or cases are unequivocally associated to predefined decisions and actions (coded in tables for recognition-primed decisions). This can be deemed intelligent depending on the number of cases in the case base and on the possibility of learning by adding new cases or adapting the table based on feedback from previous decision made or making decision by analogy using a measure of the distance between the case at hand and available recorded cases (Gilboa and Schmeidler 1995).

The second stage where intelligence can be introduced in decision support lies in the reasoning and many i-DMSS are designed to make intelligent "what-if" analysis on models and data. The principle is that of a heuristic search at different levels of representation. Here the main difficulty for designers is to complete the overall model of search by introducing the evaluation function expressing the preferences of the decision maker. When this is not possible, the decision maker must remain in the loop and the system is interactive, the decision maker expressing his or her preferences by directing the search and stopping it when they are satisfied (or they have reached a satisfying outcome, as in the concept of "bounded rationality" described by Simon).

Whether we are able to understand and model all the effects observed in human decision, the complexity lies in the difficulty to merge so much information and so many representations and models into a system. The reasoning flexibility of human beings to move freely from one level of representation to another, to capture and interpret so many images and other stimuli makes the realization of i-DMSS that support as many of the stages of the decision-making process still a challenge.

2.4 From Substantive Rationality to Bounded Rationality

Initially, Simon adopted the generally accepted definition of rationality as the matching of means to ends. This has been found to raise more problems than it solves as an individual can draw any conclusions from a false premise and that any decision relying on an erroneous diagnosis may be found to be rational in some sense. Simon (1983, p. 9-10) was aware of these problems and he stressed that a process can be rational though the facts initially considered (the diagnosis in Pomerol 1997) are false.

There has been considerable evolution in Simon's ideas between the first edition of his book and his most recent comments (1997, p. 163) in which he considerably qualified the relation between objectives and decisions by showing how the objectives and the constraints are interchangeable in the role they play in defining problems. Simon also emphasized that among all constraints, some can become objectives at a given time in the management process, and return to being constraints at other times, depending on the focus of the decision maker on one aspect or another. In an organization, an intermediate objective often becomes a means. For example, if a firm's objective is to maximize profit, then the objective 'increasing sales by 20%' may be represented as either a means or an objective. From this stems the idea of organizational chains of means and objectives (ibid. p. 83), which further complicates the evaluation of decisions. In his comments, Simon emphasizes (1997, p. 161-162) that the multilayered aspect of most decisions rules out optimization as a practical model for decision making. In DMSSs, the designer is frequently faced with this question: the constraints are a part of the model whereas the objectives depend on the decision maker and his or her evaluation. The designer will therefore have to choose between two options in dealing with each decision rule. This is an aspect of bounded rationality that Simon first referred to as "procedural rationality" (Simon 1976).

Later, Simon was led to focus on the limitations that apply to human cognition and to formulate a number of assumptions that became the foundations of what, in 1955 , he termed "bounded rationality". These can be summarized as follows:

- it is impossible to assign probabilities to all events or even simply to enumerate all possible events and their combinations;
- decision-makers' preferences are not rational in the sense of maximization of a utility function. They are in fact multicriteria and also changeable, meaning that it is impossible to spell out an overall utility function for a given choice;
- decisions spread out over time and, in organizations, form a chain in which, over time, sub-decisions are not mutually independent, but can be taken at different times and levels using nonidentical criteria; furthermore, we cannot separate preferences, actions and objectives. As Simon (1983) stated: "*the fact that sub-decisions are taken locally using partial criteria obviously – and, I would add, mathematically – counters any global optimization*" (p. 18);
- available information is fundamental and very strongly affects decisions; this is particularly evident when considering the (small) number of alternatives an individual is capable of contemplating. Attention plays a considerable role in delimiting problems and affects the subsequent decisions in that attention is a rare resource.

Simon concluded from these assumptions that managers must content themselves with suboptimal or 'satisfying' decisions. In practice, given these limitations, the decision process stops when decision makers reach a solution that satisfies them within what appears to them to be the most probable hypothesis. This notion of 'satisfying' tends to become more and more preponderant in Simon's work after 1960 as evidenced in Simon (1983) for instance. The limited rationality of 1955 gradually gives way to 'bounded rationality' (Simon 1972) and is increasingly represented in an algorithmic form already present in 1955 as the 'satisfying rule'. This algorithmic aspect highlights the sequential aspect and heuristic search nature of decision processes. This development went hand in hand with Simon's growing interest in artificial intelligence, which was explored in section 3.3.

As Simon's thinking developed, cognitive limits, with the brain as a symbol-processing system, became increasingly important elements in bounded rationality. "*In its simplest form, the theory of bounded rationality is a theory of "how to live" in an infinite world, while disposing of very limited computational means*" (Simon 1984, p. 595). Simon concluded that: "*So long as human attention is a rarer* *resource than information, we have to re-think our organizations in terms of attention management*" (Simon 1984, p. 590).

The notion of bounded rationality has had immense repercussions over the last 50 years as the first attempt at setting up a scientific framework within which the real decisions of real decision makers in real organizations could be studied against real efficiency criteria. In addition, this framework took into account the cognitive, informational and reasoning limitations of individuals and we contend that bounded rationality is a description and a representation of the way in which decisions are made in organizations. Subsequently, Simon frequently opposed procedural rationality – the rationality that takes into account the limitations of the decision maker in terms of information, cognitive capacity and attention – to substantive rationality, which is not limited to satisfying, but rather aims at fully optimized solutions.

2.5 Decisions and Organizations

Bounded rationality focuses on the individual's decision making within an organization, but in fact Simon was mainly interested in organizational decision making and the duality stemming from the fact that, while it is individuals who make decisions, it is meaningful for researchers to view organizations as having a life of their own. This duality led March and Simon (Simon 1997, p. 229) to investigate a number of key issues, including:

- the relationship between individual preferences and the objectives of the organization;
- the role and limits of authority and the hierarchy;
- channels of communication;
- departmentalization and decentralization in organizations;
- why people get involved with and remain in organizations (ibid. p. 157) (this is central in the book by March and Simon and leads to the notion of organizational equilibrium present in nearly all Simon's books);
- the role of individual psychology in constructing the company culture;
- how the above factors impact on decision-making.

Several aspects of Simon's vision are innovative. Firstly, authority and power are given their first serious definition since Dahl and Crozier's work (Jameux 1984). Authority is defined as: the ability to make decisions that engage the actions of people other than the decision maker (ibid, p. 179) and the power to arbitrate between viewpoints when there is no consensus. This led Simon to investigate the extent to which a decision made by top managers in the absence of a consensus is acceptable to subordinates and how this affects the execution of such a decision. Simon called this 'the area of acceptance'.

Simon's approach was to investigate the interaction between the organization and its structures on the one hand, and the behavior of the individual decision maker on the other hand. The idea was that the institutional setting should enable the individual decision maker to take the right decisions for the organization. Thus for

Simon, the organization provides a setting that, by various means (definition of objectives and criteria among others) affects the decision making of its members (Simon 1977, p. 51). Such ideas have since led to the notion of decentralization through objectives and, in a wider sense, of corporate culture. In connection with power, mention should be made of the interesting notion of 'uncertainty absorption' (March and Simon 1993, p. 187) that Simon defined as reasoning from previously processed facts or intermediate conclusions rather than from the facts themselves. This is a fundamental notion in the study of bureaucracies where decision makers have little contact with the real world and make little attempt to collect fresh evidence from the field.

Secondly, Simon put forward the idea of the organization as a 'role system' (Simon 1997, p. 230): the role of the organization and that of its people; and how each individual adopts their socially inherited role. Simon attempted to understand how, through role playing, the institution channels individual behavior. The role of groups and group-individual relationships are widely investigated in the book "Organizations", perhaps under the influence of March. Subsequently, this theme of social pressure on the individual was to develop in the themes of alienation and manipulation. However, unlike March, Simon always remained more of a psychologist than a sociologist in his work on organizations. It was left to Cyert and March (1963) to bring the fundamental contributions that inspired future generations to consider such problems as the avoidance of conflicts, 'the control' of avoidance of uncertainty, organizational learning and problem-driven decision (see, *e.g.*, Crozier and Friedberg 1977).

Simon borrowed from Freud the notion of identification, which he considered from two points of view: the identification of the individual with his personal role and his identification with the organization (the issues of loyalty). For Simon, the processes of identification involves the personal share that the individual has in the success of the organization, an acceptance of the philosophy of private enterprise and the value of success, and finally, a key idea that managers' decision making should be reduced to what is right for the organization (Simon 1997, p. 295). Simon's experiments (ibid. p. 296 *et seq.*) showed that on the whole, accountants formulate any organizational problem in accountancy terms, sales representatives in terms of sales, *et al.* This problem of the selective perception and interpretation of stimuli according to existing schemes is also found in pattern recognition and in the science of the artificial (Simon 1981), but it is also important in economics, in psychology and in sociology.

Throughout his work, Simon accurately anticipated, even before the World Wide Web became what it is now, the stream of information of all kinds that organizations are facing. For organizations the most critical task is not to search for or to generate still more information but to filter it so that it does not exceed their processing capacities (Simon 1977, p. 108). For managers, a similar problem arises, and hence the importance of intelligent 'interactive decision support system' filters. Even today, this serves as a justification for the field of DMSS as a key area of scientific research.

2.6 Assessing Simon's Contribution

Simon's views about postindustrial society, the utmost importance of information and the role of the manager as a decision maker have been largely confirmed and are still central in DMSS research and practice (Power 2003). Furthermore, the need for more research on the links between action and decision called for by Simon still applies today. In particular, the issue of whether action is the last step of decision and should be included in the process is still unresolved. This may mean that DMSSs should not be only deliberative but also decisive. It is our experience that most of the time they are not and that decision makers remain absolutely necessary because action (in a social framework) is also intention and commitment. Even March who is more sensitive to social aspects than Simon does not develop these ideas much. For DMSS design, this means that current systems are not sufficiently tailored for supporting action.

Today, there is wide agreement that the decision process cannot be reduced to choice (Langley *et al.* 1995), and the role of information and the building of possible alternatives are widely regarded as critical. Lewis (1991) noted that nearly 75% of authors of information system manuals adopt Simon's decision phases as a unifying thread. It is also broadly believed that managerial decision processes depend on information and the organization as well as on the individual decision maker (*e. g.* Berkeley *et al.* 1998). DMSS designers must endeavor to grasp the implications of these ideas because for a long time decision support unduly focused on the moment of choice. It is relatively recent that some DMSS and EISs address the information-gathering phase by aiding the decision maker in data miming and extracting information from databases and data warehouses, by proposing better interface designs to help managers in searching. Thus, in Power's (2003) typology, at least three of five types of DMSS focus on information and data: Data-Driven DMSS, Knowledge-Driven DMSS and Document-Driven DMSS. In addition, reviewing decision and learning from them have not truly been considered by DMSS designers and are to be found in the realm of artificial intelligence. Only experimental case-based DMSS attempt to improve their decision process through use.

The notions of heuristics search and 'satisfying' decision have been widely adopted by DMSS researchers. Interactive searches, reference point methods, local searches, *et al.*, generally invoke Simon as a source of inspiration. Interactive DMSSs, for instance, generally perform a heuristic search directed by the decision maker who progressively elicits his preferences, and stop at the first 'satisfying' decision they find (Pomerol and Adam 2003a).

Simon gained the utmost celebrity by claiming in the 1950s that computers can solve human problems and demonstrating it with such systems as GPS. The secret of problem solving is, of course, that there is no secret: when no finite algorithm can be found for a decision-making situation, the only remaining solution is to carry out a heuristic search for a desirable solution. Each heuristic search within the overall search for a solution that satisfies criteria is based on two main components: (1) evaluation, to assess the current state of the search and (2) decision, to select the most desirable goals or subgoals to move towards. The possibilities offered by such a notion, most crucially that a heuristic search provides a method to develop a path to the decision based on the decisions made by the system during the process, or by the user in the case of an interactive search, rather than following a pre-defined path, are immense and lead to more 'intelligent' systems.

On the other hand, DMSS design still faces an uphill struggle in relation to the design of possible alternatives in decision making as most DMSS treat alternatives as given and unchangeable. This is a difficult topic, mainly because alternative building follows a top-down process along the representation levels: starting with very general ideas, progressively refined towards lower level representations and towards action (Humphreys and Berkeley 1985, Lévine and Pomerol 1995, Pomerol and Adam 2003b).

A final issue deserves consideration in assessing Simon's contribution to the DMSS field: whether the very famous distinction between programmed and non-programmed decision has proven useful. We think that it could be advantageously replaced by a more easily operationalizable differentiation between automatic and interactive DMSS. It may not be very significant to say whether a DMSS addresses non programmed decisions, whereas it is clearly observable that in some cases the designer is unable to produce a complete model - especially for choice (Pomerol and Adam 2003a)-, and that the human decision maker consequently remains a key element in the process.

In section 2.4, we reviewed the key aspects of bounded rationality. There is some agreement that the behavior of human decision makers has been well captured by Simon. However, divergences occur about the applicability of this model (*e. g.* Rubinstein 1998). Bounded rationality is a fuzzy concept and as such it is not clear exactly how it should be used, given its highly subjective nature: how can an alternative be adjudged to be satisfactory by somebody other than the decision maker? The rational procedure of searching for a good alternative (heuristic search) is without doubt the most practically applicable part of the concept. Bounded rationality tells us that collective utility functions and many so-called optimizations are no more than hot air. We often hear top managers, politicians and technocrats claiming that they have made the best possible decision for the common good, a rather ambitious claim that assumes a God-like knowledge of a hypothetical collective utility function and – above all – of future events.

Simon undoubtedly thought that maximization is nonsensical, due to:

- lack of knowledge of probabilities
- multilevel, multistage, multicriteria decision process
- the fact that the preferences are not exogenous to a decision
- attention is a scarce resource

Then, it is perhaps surprising that Simon rarely refers to risk and its evaluation. Numerous experiments and much research on how people choose in risky and uncertain situations were carried out by other researchers – *e. g.* Tversky and his students and followers, who developed most of our knowledge on decisional bias (*cf.* overviews in Kahneman *et al.* 1982, von Winterfeld and Edwards 1986, Bell *et al.* 1988; Piattelli-Palmarini 1995, and Kahneman and Tversky 2000).

On the one hand, although researchers have considered the influence of individual traits in decision making, very few DMSS studies refer to decision bias and are focused on specific aids to overcome them. In particular, although it is clear that human processing of risk (*e. g.* probabilities) is very poor, very few attempts have been made to tackle the problem. In addition, DMSS are still individual and even if groupware decision has been intensively studied over the last fifteen years, we still do not see DMSS as social devices and almost no research exists on the impact of decisions and on the structure and behavior of organizations.

The second main field opened up by bounded rationality is multicriteria decision making and, more generally, the extension of operational research ('optimizing in essence') towards artificial intelligence (Simon, 1987). The use of heuristics and the so-called local methods in O.R. owe much to the impetus given by Simon and his students. A multiplicity of criteria and the resulting nonoptimization are among the features of bounded rationality that contributed to the rapid development of multi-criteria decision. The multicriteria aspect has always been present in bounded rationality, with 'partial ordering of payoffs' as a consequence (Simon 1955). This multidimensional character is the result either of having a large number of incommensurable objectives (Simon 1967), or of the fact that several individuals are involved in a group decision. This led Simon to conclude that the quest for a global optimum did not make sense. On this point Simon has had much following and multicriteria DMSSs are now commonplace.

The two remaining aspects of bounded rationality that led to further research are the question of the endogeny of the preference and the problem of limited attention. In DMSS research and practice, the former has been solved by letting the decision maker express his preference using interactive features of the system, while the latter has been addressed by developing simple, easy to handle systems rather than involved systems. This is illustrated by the shift from DMSSs with relatively sophisticated models to EIS, with few modelization and very effective displays.

Beyond bounded rationality, the impact of the work of Simon and March on sociology has been crucial. By rehabilitating the sociology of organizations and considering the interaction between the behavior of the individual and the structure of organizations and decision processes, they also totally renewed business economics, showing that a scientific approach was possible with proper experiments and measurements. The most inspired part of the work of Simon is probably his reflection on how we account for the behavior of the individual within the collective behavior of the firm.

Simon, especially in collaboration with March, paved the way towards the notion of organizational learning and all subsequent investigations in this area. Some ideas that arose from their work include the notion that the structuring of decisions leads to the creation of routinized processes and the issue of how organizations create their procedures and maintain them. What is the knowledge of an organization and how is it acquired and maintained, remain critical research questions that still fuels much debate and questioning (*cf.* Zacklad and Grundstein 2001 for recent references).

Organizational learning and knowledge management have received much attention in decision-making research (as illustrated by the themes of previous IFIP 3.3 conferences). Information systems such as data warehouses now frequently underlie DMSS building. Learning, however, is not reduced to knowledge accumulation, but also encompasses rules, procedures and routines generation. This last aspect is not well studied and although some rule-extraction methods have been proposed (statistically based or using more qualitative methods in artificial intelligence), it is still not well spread in the DMSS field.

Lastly, Simon's contribution must be examined in light of his work with Cyert and March and their book 'A Behavioral Theory of the Firm' (1963). As the title suggests, this is about reintroducing human behavior into theories of the firm. This work led to the concept of transaction cost (reflecting the fact that information and time have a price) which earned Coase the Nobel Prize (Coase, 1988), agency theory and new ideas on the concept of firm illustrated by the work of Williamson on contractual and transactional analysis in firms. Williamson analyzed the behavior of agents as decision makers with bounded rationality. This led him to discuss the firm as opposed to the market in terms of adaptation to changes in the environment and speed of reaching a decision (Williamson 1991).

2.7 Concluding Comments

We can only conclude, perhaps predictably, that, Simon's legacy is considerable. In the fields we considered in this chapter, Simon was a forerunner whose work is still central today. His influence has been direct on research in decision-making processes, especially in terms of refocusing research efforts on the early stages of intelligence and design. He has also directly influenced our thinking in terms of satisfying and bounded rationality and he has codifed the idea of problem structuredness. He has been one of the founders of artificial intelligence and has indirectly influenced research on intelligent agents, on multicriteria decision making and on knowledge management. However, we further argue that his ideas have not only led to substantial advances in a number of key domains of research, but even have the potential to deliver further radical progress in the future as the full potential of Simon's work is understood. This seems particularly true when considering intelligent DMSS, where Simon's ideas related to decision by intuition and how to mimic this type of decision making in case-based DMSSs have not been pursued to any great extent. The linkage between decision and action (ie the inclusion of support for action in DMSS design) has also yet to be investigated.

Simon's ability to collaborate with others and his gift for communication, borne out by the large number of coauthored papers - over 80 in all (Simon 1991, p.387)-, made him a great source of inspiration for many and ensure his place in posterity. It is worth noting that despite the uncertain aspects of the fields of management and decision (particularly as they were in the 1940s and 1950s), Simon always managed to stick to facts and, in his own words, to accept nothing that cannot be upheld by the facts. Though a champion of bounded rationality, Simon was a rationalist himself. As March puts it (quoted in Weil 2000, p. 35): "*Herbert Simon is an unrepentant knight of the enlightenment. Not Freud, but Descartes [....] He studies the limitations of reason in the name of reason*". Bounded rationality is not the absence of rationality, but rather the question of what one can do that is rational without full knowledge of the facts. This is why Simon is often described as 'an enlightened man'. Certainly an enlightened man, but also a true scientist.

References

Bell ED, Raiffa H, Tversky A (eds.), (1988) Decision-making, Cambridge University Press, Cambridge, USA.
Berkeley D, Widmeyer G., Brezillon P, Rajkovick V (1998) Context Sensitive Decision Support Systems, Chapman and Hall, London, UK.
Bonczek RH, Holsapple CW, Whinston AB (1981) Foundations of Decision Support Systems, Academic Press, NY.
Coase RH (1988) The Firm, the Market and the Law, The University of Chicago Press, Chicago, USA.
Crozier M, Friedberg E (1977) L'Acteur et le Système, Editions du Seuil, Paris.
Cyert RM, March JG, (1963) A Behavorial Theory of the Firm, Prentice-Hall, Englewood Cliffs, N.J.
Damasio AR, Damasio H, Christen Y (eds) (1996) Neurobiology of Decision-making, Springer, Berlin.
Gilboa I, Schmeidler D (1995) Case-based decision theory, Quarterly Journal of Economics 110; 605-639.
Gilboa I, Schmeidler D (2000a) Case-based knowledge and induction. IEEE Transactions on Systems, Man and Cyberne 30: 85-95.
Gilboa I, Schmeidler D, (2000b) Cognitive Foundations of Inductive Inference and Probability : An Axiomatic Approach, Working Paper, Tel Aviv and Ohio State Universities.
Humphrey P, Bekerley D, (1985) Handling uncertainty: levels of analysis of decision problems, in G. Wright (Ed.) Behavioural Decision-making, Plenum Press, London.
Jameux C (1984) Pouvoir, information et coordination : les contributions de H.A. Simon au concept de pouvoir, in Colloque de la Grande Motte, A. Demailly et J.L. Le Moigne (Eds.), 336-345.
Janis I (1972) Victims of Groupthink, Houghton Mifflin Comp, USA.
Kahneman D, Tversky A (eds.) (2000) Choices, Values and Frames, Cambridge University Press, Cambridge UK.
Kahneman D, Slovic P, Tversky A (eds.) (1982) Judgment under Uncertainty: Heuristics and Biases, Cambridge University Press, Cambridge, U.K.
Klein GA (1993) A recognition-primed decision (RPD) model of rapid decision-making, in Decision-making in Action, Models and Methods, G.A. Klein, J. Orasanu, R. Calderwood, C.E. Zsambok (Eds.), Ablex, Nordwood N.J., 138-147.
Langley A, Mintzberg H, Pitcher P, Posada E., Saint-Macary, J (1995) Opening Up Decision-making: the View from the Black Stool, Organisation Science, 6, 3, 190-205.
Lévine P, Pomerol JC (1989) Systèmes Interactifs d'aide à la Décision et Systèmes Experts, Hermès, Paris.
Lévine P, Pomerol JC (1995) The role of the decision maker in DSSs and representation levels, in Proceedings of the 29 th Hawaï International conference on System Sciences, volume 3, Nunamaker Jr. J. F. and Sprague R. H. (Eds), IEEE, 42-51
Lewin K, Dembo T, Festinger L, Sears P (1944) Level of aspiration, in Personality and the Behavior disorders, J.M. Hunts (Ed.), Ronald Press, NY, 333-378.
Lewis PJ (1991) The decision-making basis for information systems : contribution of Vicker's concept of appreciation perspective, European Journal of Information Systems 1: 33-43.
March JG, Simon HA (1993) Organizations (second edition; first edition 1958), Basil Backwell, Oxford, U.K.
Mintzberg H (1990) Le Management, Voyage au Centre des Organisations, Les Editions d'Organisations, Paris.
von Neumann J, Morgenstern O (1944) Theory of Games and Economic Behavior, Princeton University Press, N.J.

Newell A, Simon H (1963) GPS: a program that simulates human thought, in Computer and Thoughts, Feigenbaum, E. and Fieldman, J. (Eds), McGraw-Hill.
Newell A, Simon HA (1972) Human Problem Solving, Prentice-Hall, Englewood Cliffs.
Newell A, Simon HA (1976) Computer science as empirical inquiry: symbols and search. Communications of the ACM, 19(3) : 113-126.
Piattelli-Dalmarini M (1995) La réforme du Jugement ou comment ne plus se tromper, Odile Jacob, Paris.
Pomerol JC (1994) Le monde de l'aide à la décision, LAFORIA - Working Papers Series, Number 94 / 20, 1-26.
Pomerol JC (1997) Artificial intelligence and human decision-making, European Journal of Operational Research 99: 3-25.
Pomerol JC (2001) Scenario development and practical decision-making under uncertainty, Decision Support Systems 31: 197-204.
Pomerol JC (2005) La Décision Humaine : Reconnaissance plus raisonnement, in concepts et méthodes pour l'aide à la décision, D. Bouyssou, D. Dubois, M. Pirlot and H. Prade (Eds), Hermès, forthcoming.
Pomerol JC, Adam F (2003a) From human decision-making to DMSS architecture, in M. Mora, G.A. Forgionne and J.N.D. Gupta, Decision-making Support Systems, Idea Group, Hershey PA, 40-70
Pomerol JC., Adam F (2003b) Multi-level and multi-model DSSs, in DSS in the uncertainty of the internet age, T. Bui, H. Sroka, S. Stanek et J. Goluchowski (Eds.), Karol Adamiecki University of Economics Press, Katowice, Poland, 333-346.
Pomerol JC, Adam F (2005) Decision-making support systems and representation Levels, Encyclopedia of Information Science and Technology, M. Khosrow-Pour (Ed.), Volume IV, 766-771.
Power JD (2003) Defining decision support constructs, in DSS in the uncertainty of the internet age, T. Bui, H. Sroka, S. Stanek et J. Goluchowski (Eds.), Karol Adamiecki University of Economics Press, Katowice, Poland, 51-61.
Rubinstein A (1998) Modeling Bounded Rationality, MIT Press, MA.
Selten R (2002) What is bounded rationality, in Bounded Rationality : the adaptive toolbox, Gigerenzer G. and Selten R. (Eds.), MIT Press, MA, 13-36.
Simon HA (1955) A Behavioral model of rational choice, Quarterly Journal of Economics, 69, 99-118.
Simon HA., (1967) Motivational and Emotional Controls of Cognition, Psychological Review, 74, 29-39.
Simon HA (1972) Theories of bounded rationality, in Decision and Organization, C.B. McGuire and R. Radner (Eds.), North Holland, Amsterdam, 161-176.
Simon HA (1976) From Substantive to Procedural Rationality, in Methods and Appraisal in Economics, S.J. Latsis (Ed.), Cambridge University Press, N.Y., 129-148.
Simon HA (1977) The New Science of Management Decision (3rd revised edition; first edition 1960) Prentice-Hall, Englewood Cliffs, NJ.
Simon HA (1979) Models of Thought, Yale University Press, New Haven.
Simon HA (1981) The Sciences of the Artificial (2nd. edition; 1st. edition 1969) MIT Press, Cambridge, Mass.
Simon HA (1983) Reason in Human Affairs, Basil Blackwell, Oxford, U.K.
Simon HA (1984) Commentaires dans le Cadre du Colloque de la Grande Motte, in Demailly A. and Le Moigne J.L., 577-619.
Simon HA (1987) Two heads are better than one: the collaboration between AI and OR, Interfaces 17: 8-15.
Simon HA (1991) Models of my Life, BasicBooks.
Simon HA (1995a) Explaining the ineffable: AI on the topics of intuition, insight and inspiration, Proceedings of IJCAI-95, 939-948.

Simon HA (1995b) Artificial Intelligence: an empirical science. Artificial Intelligence, 77(1) : 95-127.

Simon HA (1997) Administrative Behavior (4th expanded edition; first edition 1947), The Free Press, N.Y.

Sprague RH, Carlsson ED (1982) Building Effective Decision Support Systems, Prentice Hall, NJ.

Taylor FW (1911) The Principles of Scientific Management, Harper and Bres, New-York.

Weil T (2000) Invitation à la lecture de James March, Presses de l'Ecole des Mines de Paris.

Williamson OE (1991) The Logic of Economic Organizations, in The Nature of the Firm, Williamson O.E. and Winter S.G. (Eds.), Oxford University Press, Oxford.

von Winterfeldt D, Edwards W (1986) Decision analysis and Behavorial Research, Cambridge University Press, Mass.

Zacklad M, Grundstein M (eds.) (2001) Ingénierie et Capitalisation des Connaissances (Vol. 1); Management des Connaissances (Vol. 2), Hermès Science, Paris.

3

Synergizing the Artificial Intelligence and Decision Support Research Streams: Over a Decade of Progress with New Challenges on the Horizon

Jason Nichols and Michael Goul

Department of Information Systems, W.P. Carey School of Business, Arizona State University, P.O. Box 874606 Tempe, AZ 85287-4606, USA,
e-mails: Jason.Nichols@asu.edu, Michael.Goul@asu.edu

Over the past two decades, the convergence of artificial intelligence and decision support technologies has driven individual, team, and organizational computing research and practice in exciting new directions. In 1992, Goul, *et al.* examined the patterns of a decade of progress and developed propositions regarding the direction and impact of intelligent decision support systems. A decade later, the propositions from 1992 have proven an insightful roadmap for breakthroughs in the individual, team, and organizational streams of intelligent decision support research. In this chapter, we reflect on the patterns of progress over the last ten years, compare them with the propositions put forth in 1992, and offer new perspectives based on an emerging stream in the literature base: intelligent interorganizational decision support. Interorganizational support represents the next generation of intelligent decision support, and the new domain brings with it new challenges and opportunities. We look to patterns in both the maturing streams of individual, team, and organizational intelligent decision support, as well as those emerging in the new interorganizational stream, to propose direction for and identify significant challenges to be addressed in future work as intelligent decision support research enters its third decade.

3.1 Introduction

In a special issue of *Decision Sciences* they coedited in 1992, Goul, Henderson, and Tonge examined the debate regarding whether Artificial Intelligence (AI) can and should serve as a reference discipline for decision support system (DSS) research (Goul *et al.* 1992). Their examination culminated in a fundamental proposition (the "knowledge-based DSS" proposition) regarding the scope and course of the evolution of DSS/AI as the fields matured from the information to the knowledge era[1]. Special issue papers served as the primary evidence of the potential evolutions they envisioned. In the following, several of the specific propositions are discussed in order to provide foundation for a more formal discussion of progress over the years. At the most basic level, their view was that in the knowledge era, AI-enabled decision support systems shift in capabilities from passive information gathering and presentation to take on a role of *decision maker* or *decision-making agent* as a result of embedded knowledge that codifies the decision process - where possible -.

The knowledge-based DSS proposition has implications for decision support at the individual, group, and organization levels. Philosophically, the adoption of knowledge to transform DSS systems into true decision-making agents evolves individual DSS processes to *de facto* group processes as decisions become a collaborative activity between the individual and the agent. At the group level then, group decision support system (GDSS) research evolution in the knowledge era requires the expansion of the decision-maker landscape from geographically and temporally dispersed human constituents to include interfaces with knowledge-based, automated decision-making agents. Within the context of a common platform for interaction provided by the GDSS architecture, the line between human and automated decision-making agents becomes blurred.

The knowledge-based DSS proposition examines DSS research at the organizational level as well. Embedded knowledge in DSS has the capability to redefine an organization - right down to its fundamental structure -. The rationality of a knowledge-enabled decision-making agent within its domain is bounded only by the limits of the hardware upon which it operates (size, speed, *et al.*). In contrast to human decision makers operating with passive support systems, the expansion of rationality afforded by knowledge-enabled agents allows for a more malleable governance structure. In other words, decision-making agents presented with codified control information and embedded with knowledge of decision-making processes can flatten governance hierarchies due to their ability to interpret and react to greater volumes of information. This results in the ability to shift the centralization of process knowledge from departments to systems, allowing for a decentralized and efficient application of such processes across departments – a perspective that would otherwise lack independent, higher-level process understanding. Further, the expanded bounds of agent rationality over human limits can be applied for a finer level of control information detail that would otherwise overwhelm the decision-making process. The level of granularity for information that drives decisions can then be adjusted to more closely fit the process rather than to fit the limits of the decision maker.

Goul, *et al.* found supporting evidence that research in individual DSS is adopting AI techniques to support knowledge-based automated decision making at an increasing rate. However, in 1992, the implications of AI's adoption as a reference discipline for group and organizational DSS remained open questions. This chapter looks to revisit these questions to uncover the effects of AI adoption in

---

[1] Benjamin and Scott-Morton [Benjamin, R. and Scott-Morton, M. (1988) *Interfaces,* 18**,** 81.consider the transition from the information era to the knowledge era as characterized by the ability to "capture qualitative knowledge and exploit it with new forms of systems architecture".

the knowledge era with respect to individual, group, and organizational DSS research. Evidence for AI's impact on decision support research was substantiated through a review of *Decision Support Systems* – the journal - spanning the last decade. *Decision Support Systems* provides a solid basis for sampling decision support research as it has evolved over time into the preferred journal of DSS scholars and DSS interest groups alike. This is evidenced by both the caliber of researchers publishing in the journal as well as special interest group (SIG) DSS-sponsored special issues finding a home there, reflecting the preferences of the SIG DSS community.

It becomes clear through this review that a fourth category, interorganizational DSS, is becoming prominent within the research stream. The impact of artificial intelligence on this new area of DSS research will be explored here as well, and the knowledge-based proposition from 1992 will be updated to include implications and observations for crossenterprise intelligent decision support research. The interorganizational DSS category represents the next major challenge on the horizon for furthering research-stream synergies.

3.2 AI-enabled DSS for the Individual

Goul, *et al.* 1992 proposed and supported the observation that DSS research has moved past a debate of differences between DSS and AI and is now able to focus on how each field can be leveraged in an integrated research program. Since 1992, support for this observation has advanced from the early stages of AI integration into decision support characterized by the use of expert systems, neural networks, and other core AI techniques to include a layer of decision support abstraction that envelopes the capture and application of user preferences as a context for intelligent decision making (Chen and Lin 2003, Mayer 1998, Palma-dos-Reis and Zahedi 1999, Yang and Chung 2004). Further levels of abstraction that demonstrate integration of AI and DSS in the knowledge era include Xiang and Poh's (Xiang and Poh 2002) work that looks at model construction in decision support systems as a knowledge based procedure and considers the tradeoff between model quality and model tractability in time-constrained decision making, and Hess *et al.* (2000) who consider the role of autonomous agents in a DSS as one of a "personal computing assistant" providing support for interoperability (integration of heterogeneous applications and networks) and user interfaces (personalized information filtering). While Goul, *et al.* (1992) was able to find ample support for the knowledge based proposition at the individual level, research in this subfield over the past decade has demonstrated an exciting degree of growth from the technical roots of AI implementation within the individual decision support framework to higher levels of abstraction that consider the impact of artificial intelligence on customization, timing, and interoperability.

3.3 AI-enabled DSS for the Group

From a review of *Decision Support Systems* – the journal - over the past decade, few group support system articles can be found, and even fewer that leverage artificial intelligence as a reference discipline. Exceptions include Pinson *et al.* (1997) where collaborative strategic planning is enhanced by a distributed agent-based decision support system that embeds strategic and domain knowledge across the agents themselves who collaborate with users through the use of blackboards and message passing. Also, Talukdar (1999) considers the theory surrounding effective collaboration of autonomous agents in a decision support context.

Perhaps the lack of evidence for artificial intelligence as a reference to group support system research has to do with collaborative computing as a discipline embracing the group support concept. Collaborative computing historically targets communications issues in the absence of decision models as can be found in DSS research. In 1992 it was predicted that machine-based intelligent agents will interface with human users through the common platform of the group decision support system in order to bring to bear specialized knowledge, skills, and experience. While the premise is still valid considering the few papers found in *Decision Support Systems*, it appears that artificial intelligence has had a greater impact on DSS research at the organizational level.

3.4 AI-enabled DSS for the Organization

Goul, *et al.* 1992 observed that the knowledge-based proposition suggests the infusion of knowledge within organizational decision support systems (ODSS). This infusion, if process-centric in nature, will enable the circumvention of traditional bureaucratic procedures that arise as a function of the organizational structure itself. As a result, ODSS should result in simplicity of processes and lower operational costs. However, at the time of that publication, little work had been done in the organizational subfield of AI/DSS.

Over a decade later, the organizational subfield has flourished in a pattern similar to that predicted in 1992. A number of papers have targeted knowledge infusion in ODSS from the perspective of knowledge capture, knowledge access, and metrics. Balasubramanian *et al.* (1999) presents a goal-oriented modeling schema for capturing and organizing knowledge in the decision-making process that can be extended by decision model preservation via model marts and warehouses as suggested by Bolloju (2002). Papers such as Courtney (2001) have investigated the leveraging of captured decision support knowledge via unbounded systems thinking in order to reorient ODSS towards "wicked" decisions. As a response to heightened focus on knowledge-enabled ODSS, Nemati *et al.* (2002) proposes in their knowledge-warehouse architecture that the effectiveness of a DSS will eventually be measured by its ability to promote and enhance knowledge and how well it improves mental models and understanding of decision makers - and therefore their decision making -.

ODSS research has, as predicted in 1992, looked at process-specific knowledge as a tool to simplify a knowledge worker's procedures. In Azoulay-Schwartz *et al.* (2004), the purchasing function benefits from an agent-based ODSS that compares supplier offerings in the context of knowledge regarding prior supplier performance in order to simplify the decision regarding the cost/quality tradeoff. Strategic management also benefits from a convergence of process knowledge with business intelligence in Ali and Wallace (1997) where strategic objectives are embedded into data-mining algorithms that are then able to adapt and find proper strategic performance measures. Similar advances in process-specific knowledge-enabled decision support have considered both marketing (Curry and Moutinho 1994) and project management (Garcia *et al.* 2004).

One implication of the knowledge-based proposition within the ODSS context is that organizational structures can become more malleable and have the potential to diminish in formality as the process complexity for intrafirm decision making is offloaded to the ODSS itself. Evidence of this can be seen at both the tactical and strategic levels. Pricing and technician assignment, two tactical areas that typically rely on hierarchal governance structures for decision making, are shown to be good candidates for AI-enabled decision-making based on knowledge-infused decision models as found in Sung and Lee (2000) and Lazarov and Shoval (2002) respectively. Evidence that these two areas perform as well as or better than under an AI/DSS enabled platform suggests that the historical governance structure for decision making in these domains is no longer required.

Information required to drive strategic decision making in an organization typically requires engagements between departments (*i. e.* the information required is crossdepartmental), which may at times prove to be untimely and expensive. In Liu and Lu (2003) a multiagent system is proposed to monitor critical success factors, the output of which can inform the strategic decision-making process while circumventing continual crossdepartment engagement for monitoring purposes. In Houben *et al.* (1999), an expert-systems approach is taken to streamline SWOT analysis through system-based identification of internal strengths and weaknesses.

Another implication of the knowledge-based proposition is that rationality of decision makers at the organization level is increased. This is to say that decision makers can benefit from ODSS in the knowledge era because the ODSS itself can consider greater information at greater depths than can a decision maker acting alone. The result is a decision that explores more of the problem space in less time. Examples include Ozbayrak and Bell (2003) where a knowledge-enabled expert system handles changes in scheduling, availability, and faults for a flexible manufacturing system and Park and Park (2003) where merchandise management is offloaded to an ODSS that simultaneously applies multiple decision models to the problem of merchandise levels over time.

A newer development in organizational decision support that augments the 1992's predictions is research targeting the interoperability and coordination of decision tools and data. This is called for in Santhanam *et al.* (2000) and addressed in Sen (2004) where the development of a metadata warehouse to manage and maintain integration details is proposed. From this perspective, it can be observed that research in integration of DSS components will rejuvenate the interest in organizational structure flexibility and reduction in crossdepartmental engagement complexity. This complexity is derived from a "service culture" now pervasive in the management philosophies of modern organizations - including in those units typically assigned responsibility for elements of enterprise computing (EC) projects. New, highly complex EC projects require an upfront configuration phase to assess the needed level of engagement from each unit in order to produce a robust and effective solution (Cameron 2002). This configuration phase results in an "engagement model" that involves some or all of those service units. When each service unit ascribes to its discipline-based focus, the resulting engagement model requires significant integration overhead due to lack of common vocabulary, alternative perspectives of the problem domain, lack of a common understanding of each discipline's toolsets being brought to bear in the analysis of that problem domain, *et al.* In addition, the personnel who are typically self-taught to become effective interservice unit integrators often come to be in such high demand that they bottleneck the efficient and cost-effective delivery of a portfolio of ongoing projects. Therefore, once the complexity of system interaction becomes significantly low, the bottleneck to seamless resource configuration and use will be organizational rather than technical.

3.5 AI-enabled Inter-organizational Decision Support

Since 1992, the landscape of business has shifted from an emphasis on vertical integration to an emphasis on core competencies and virtual integration across a set of firms comprising a value-adding supply chain or network. The distributed nature of supply chains impacts the ability for any individual firm to access information outside of their own locus of control. This drives a need for decision support that spans outside traditional firm boundaries to encompass those firms connected and collaborating to produce goods and services as noted in Shim *et al.* (2002).

Bui and Lee (1999) lay out a framework for agent-based DSS as a process of developing a coordinated workflow of collaborating agents that are able to support the problem-solving process across the complete set of coordinating firms. This framework is augmented in Chang and Lee (2004) by the proposal of a formal model request language and model selection and optimization methods for interorganizational model-agent based coordination. Examples of interorganizational decision support systems can be found in Hess *et al.* (2000), Karacapilidis and Moraitis (2001), Kimbrough *et al.* (2002), and Liang and Huang (2000).

The independence of firms across a supply chain presents an additional dimension of complexity to interorganizational decision support. Some issues that arise from this include an inability to integrate models and data across firms, a need to provide levels of process model abstraction in order to preserve proprietary interests of individual firms, and incentive misalignment for pursuing globally optimal decisions (growing total value in the supply chain) versus those that are locally optimal to each firm (expanding an individual firm's share of the supply-chain's value).

The issue of model and data integration has been considered from both the perspective of hierarchical data composition using a filter space approach (Chari, 2003) as well as a metadata approach that describes DSS-relevant content both internal and external to the organization (Gregg *et al.* 2002). The management of both filter spaces and metadata, however, is unique to each individual supply-chain configuration and requires rework should the network of participating firms change. This serves to constrain the velocity at which supply chains can reconfigure themselves and creates a barrier to supply-chain automation in the sense of intelligent automated reconfiguration to adapt to real-time perturbations in the network.

Model granularity control is a requirement in interorganizational decision support systems in any instance where the decision process at one firm requires model information at another firm that is tied to a detailed and proprietary process and/or decision model. The use of activity/process states leveraging process views is proposed in Liu (2004) as a mechanism to abstract internal processes from the interface presented to external firms in the interorganizational decision support framework.

Incentive misalignment across the supply chain with respect to decision support appears to be an open issue that can benefit from the economics literature on incentive design. Early work in this area includes Weigand and van den Heuvel (2002) that proposes a formal xml-based contract-specification language that can be leveraged to provide incentive through contracting at the level of the decision support system itself. This method has appeal due to its instantiation at the system level that provides an opportunity for AI/DSS research to consider the implications of automated contract negotiation for participation in decision processes within the supply chain. The potential for a secondary market for data and models within a supply chain and the impact of this market on traditional roles in the supply chain is an interesting extrapolation from the dynamics of an interorganizational decision support network.

3.6 Proposition Revisited

In 1992 it was observed that decision support research at the individual level would move past a debate of AI as a contributing reference discipline to engage AI directly in a research stream that melds AI and DSS together. This is confirmed over the past decade of AI/DSS research, and furthered by the observation that *AI/DSS research at the individual level is beginning to engage at a higher level of abstraction from technical issues that suggests that the subfield is reaching maturity.*

The organizational-level observation from 1992 predicted that knowledge embedded in ODSS systems would circumvent conventional bureaucratic procedures that were a function of traditional organizational structures. A review of the literature suggests that this prediction is supported, but also opens up a new issue that may drive the focus of organizational DSS research; namely engagement complexity. AI/DSS research in this subfield has simplified the technical relationship between a knowledge worker and the processes and data required for that knowledge worker to perform a given task. However, an artifact of traditional organizational structures persists in the form of engagement complexity for separate organizational entities to interact with each other. The cost and complexity of crossdepartmental collaboration that can be attributed to the policies and procedures implemented regarding crossdepartmental engagement may begin to serve as the bottleneck for ODSS since the technical cost and complexity continues to be reduced thanks to AI/DSS research contributions in the subfield. Therefore, *the reality of seamless ODSS requires future investigation of a less technical nature that can address the business process cost, rigidity, and complexity of interdepartmental engagement models.*

In 1992, little was said and/or done by way of interorganizational decision support. However, the foundations for a value proposition of this type of decision support can be mapped from the role of artificial intelligence in operations management as observed by H. A. Simon and quoted in Goul (1992). In summary, Simon suggested that the infusion of operations management with artificial intelligence would open up the potential to tackle the complete set of organizational decision making. In a contemporary sense, operations management must be viewed in the context of the supply chain, but the fundamental tenet of Simon holds true. AI/DSS research in interorganizational decision support systems requires a mapping of organizational DSS concepts and theory to an environment of greater complexity. Early work in this area over the past decade has shown promising technical progress. However, a focus on the AI-enabled operations management literature within the AI/DSS research stream can help to organize the compliments of the two disciplines. The union of these two fields is a natural one that can leverage the common reference discipline of artificial intelligence as a bridge for collaboration. As an example, AI/OM research is beginning to look at agent based supply chain management simulation and the decomposition of such a framework into control elements, interaction protocols, and agent typologies (Swaminathan *et al.* 1998, Fox *et al.* 2000). The output of AI/OM research of this sort includes platforms for simulation of configurations within a supply chain that can be leveraged for business process redesign.

AI/DSS research can contribute to this stream by way of expanding the framework from simulation agents that emulate supply-chain participants to distributed agents that act on behalf of the supply chain participants themselves. Requirements for this type of research fusion include: 1) introduction of DSS interorganizational research to manage the application of data and models in a heterogeneous environment, 2) a common process modeling language to support dynamic reconfiguration with minimal model management overhead, and 3) standards for communication (look up and discovery) of process models and supply chain partners. Interestingly, all of the pieces to this puzzle exist in separate research areas. The fusion of DSS and operations management research can be facilitated by the shared reference discipline of Artificial Intelligence. The Unified Enterprise Modeling Language (UEML), now heading towards its second version, seeks to mitigate the requirement for model merging through transformation or middleware by providing a language that can be used commonly across diverse organizations to represent the characteristics of their enterprise. A common modeling language such as UEML is attractive because it seeks to mitigate continued middleware development for decision support when entities in the supply chain are replaced or when new entities enter. Communications platforms such as the UDDI layer of the Web Services protocol stack can be leveraged for dynamic lookup and discovery of supply chain participants where enterprise models are shared via web services. In the knowledge era, metadata can be mapped to enterprise models within the supply chain in order for participants to monitor contractual terms such as performance and to identify and respond to anomalies in crossenterprise processes. In this sense, the promise of AI-enabled operations management augments the promise of AI-enabled decision support at the inter-organizational level and the result is seemingly greater than the sum of its parts.

It can therefore be observed that *1) the importance of common modeling techniques will increase as research in interorganizational decision support continues, 2) operations management and decision support will converge in their use of artificial intelligence as a reference discipline, and 3) organizational issues such as incentive alignment and model abstraction must be addressed in order for the complete vision of the automated supply chain as outlined to be achievable.*

3.7 Conclusion

As a reference discipline, the application of Artificial Intelligence has flourished in decision support system research over the past decade. The 1992's proposition in Goul *et al.* (1992) that AI can broaden DSS research by selectively incorporating machine-based expertise is substantially supported in the individual and organizational levels as well as in a newer domain for decision support: the interorganizational level.

Moving forward, it is observed that continued interorganizational DSS research will merge with operations management research streams that leverage artificial intelligence as a reference discipline as well, the importance of common modeling languages has increased due to the heterogeneity of the interorganizational landscape, and common communications protocols such as Web Services can serve as a backbone for dynamic interorganizational process reconfiguration.

At both the organizational and interorganizational level, continued research will reach a level of maturity in the field where organizational issues become principal to the success of AI-enabled DSS artifacts. At the organizational level, engagement cost and complexity across departments will become the bottleneck to an otherwise flexible decision support architecture. At the interorganizational level, incentives for optimal global decision making and mechanisms for abstraction of proprietary information from shared process models will become critical as further design science efforts solidify the technical vision of the automated supply chain.

References

Ali F, Wallace W (1997) Bridging the gap between business objectives and parameters of data mining algorithms. Decision Support Systems, 21(1): 3-15

Azoulay-Schwartz, R, Kraus S, Wilkenfeld J (2004) Exploitation vs. exploration: choosing a supplier in an environment of incomplete information. Decision Support Systems, 38(1): 1-18

Balasubramanian P, Nochur K, Henderson J, Kwan M (1999) Managing process knowledge for decision support. Decision Support Systems, 27(1): 145-162

Benjamin R, Scott Morton M (1988) Information technology, integration, and organizational change. Interfaces, 18(3): 86-98

Bolloju N, Khalifa M, Turban E, (2002) Integrating knowledge management into enterprise environments for the next generation decision support. Decision Support Systems, 33(2): 163-176

Bui T, Lee J (1999) An agent-based framework for building decision support systems. Decision Support Systems, 25(3): 225-237

Cameron J (2002) Configurable Development Processes. Communications of the ACM,, 45(3): 72-77

Chang Y, Lee J (2004) Case-based modification for optimization agents: AGENT-OPT. Decision Support Systems, 36(4): 355-370

Chari K (2003) Model composition in a distributed environment. Decision Support Systems, 35(3): 399-413

Chen J, Lin S (2003) An interactive neural network-based approach for solving multiple critera decision-making problems. Decision Support Systems, 36(2): 137-146

Courtney J (2001) Decision-making and knowledge management in inquiring organizations: toward a new decision-making paradigm for DSS. Decision Support Systems, 31(1): 17-38

Curry B, Moutinho L (1994) Intelligent computer models for marketing decisions. Management Decision, 32(4): 30-35

Fox MS, Barbuceanu M, Teigen R (2000) Agent-oriented supply-chain management. International Journal of Flexible Manufacturing Systems, 12(2,3): 165-188

Garcia M, Quintales M, Garcia F, Polo J (2004) Building knowledge discovery-driven models of decision support in project management. Decision Support Systems, 38(2): 305-317

Goul M, Henderson J, Tonge F (1992) The emergence of artificial intelligence as a reference discipline for decision support systems. Decision Sciences, 11(2): 1273-1276

Gregg D, Goul M, Philippakis A (2002) Distributing decision support systems on the WWW: the verification of a DSS metadata model. Decision Support Systems, 32(3): 233-245

Hess TJ, Rees LP, Rakes TR (2000) Using autonomous software agents to create next generation of decision support systems. Decision Sciences, 31(1): 1-31

Houben G, Kenie K, Vanhoof K (1999) A knowledge-based SWOT-analysis system as an instrument for strategic planning in small and medium sized enterprises. Decision Support Systems, 26(2): 125-135

Karacapilidis, N, Moraitis P (2001) Building an agent-mediated electronic commerce system with decision analysis features. Decision Support Systems, 32(1): 53-69

Lazarov A, Shoval P (2002) A rule-based system for automatic assignment of technicians to service faults. Decision Support Systems, 32(4): 343-360

Liang T, Huang J (2000) A framework for applying intelligent agents to support electronic trading. Decision Support Systems, 28(4): 305-317

Liu D (2004) Business-to-business workflow interoperation based on process-views. Decision Support Systems, 38(3): 399-419

Kimbrough S, Wu D, Zhong F (2002) Computers play the beer game: can artificial agents manage supply chains? Decision Support Systems, 33(2): 323.

Liu R, Lu Y (2003) Distributed agents for cost-effective monitoring of critical success factors. Decision Support Systems, 35(3): 353-366

Mayer M (1998) Future trends in model management systems: parallel and distributed extensions. Decision Support Systems, 22(4): 325-335

Nemati H, Steiger D, Iyer L, Herschel R (2002) Knowledge warehouse: an architecture integration of knowledge management, decision support, Artificial Intelligence and data warehousing. Decision Support Systems, 33(2): 143-161

Ozbayrak M, Bell R (2003) A knowledge-based decision support system for the management of parts and tools in FMS. Decision Support Systems, 35(4): p. 487-515

Palma-dos-Reis A, Zahedi F (1999) Designing personalized intelligent financial decision support systems. Decision Support Systems, 26(1): 31-47

Park J, Park S (2003) Agent-based merchandise management in Business-to-Business Electronic Commerce. Decision Support Systems, 35(3): 311-333

Pinson S, Anacleto L, Moraitis P (1997) A distributed decision support system for strategic planning. Decision Support Systems, 20(1): 35-51

Santhanam R, Guimaraes T, George J (2000) An empirical investigation of ODSS impact on individuals and organizations. Decision Support Systems, 31(1): 51-72

Sen A (2004) Metadata management: past, present and future. Decision Support Systems,, 37(1): 151-173

Swaminathan J, Smith S, Sadeh N (1998) Modeling Supply Chain Dynamics: A Multiagent Approach. Decision Sciences, 29(3): 607-632

Shim JP, Warkentin M, Courtney JF, Power DJ, Sharda R, Carlsson C(2002) Past, present, and future of decision support technology. Decision Support Systems, 33(2): 111-126

Sung N, Lee J (2000) Knowledge assisted dynamic pricing for large-scale retailers. Decision Support Systems, 28(4): 347-363

Talukdar S (1999) Collaboration rules for autonomous software agents. Decision Support Systems, 24(3): 269-278

Weigand H, van den Heuvel W (2002) Cross-organizational workflow integration using contracts. Decision Support Systems, 33(3): 247-265

Xiang YP, Poh K (2002) Knowledge-based time-critical dynamic decision modelling. The Journal of the Operational Research Society, 53(1): 79-87

Yang C, Chung A (2004) Intelligent infomediary for web financial information. Decision Support Systems 38(1): 65-80

4

From Knowledge Discovery to Computational Intelligence: A Framework for Intelligent Decision Support Systems

Sergio Viademonte and Frada Burstein

School of Information Management & Systems, Monash University, Caulfield Campus,
26 Sir John Monash Drive, Caulfield East, Victoria, Australia,
e-mails: sergio.viademonte;frada.burstein@infotech.monash.edu.au.

The research described in this chapter is concerned with investigating the combination of knowledge discovery in database and intelligent computing technologies, in developing a framework for intelligent decision support systems (IDSS). In this context, the chapter presents an approach for IDSS through the combination of data mining (DM) technology with artificial neural networks (NN) in a hybrid architecture called the DM-NN model. This research draws from the concepts of computational intelligence, knowledge discovery in databases and decision support.

4.1 Introduction

Intelligent decision support systems are expected to incorporate specific domain knowledge and perform some type of intelligent behaviour, such as learning and reasoning, in order to support decision-making. The need to incorporate domain knowledge and intelligent capabilities in decision support systems has been identified in various forms and models by many researchers (Simon 1977, Sprague 1993, Turban *et al.* 2005)

For example, Teng *et al.,* (1988) and Turban and Aronson (1998) proposed an architecture for an IDSS in which a knowledge acquisition subsystem is linked to an intelligent supervisor, which is implemented through an inference engine. Another example is a framework proposed by Burstein *et al.* (1998) to support decision making by combining a case base, a database, and a rule base into an intelligent advisory system. The proposed framework is built around a collection of organizational knowledge to make it accessible to decision makers. Shim *et al,* (2002) relate to the concept of model-based decision support, which concerns systems for decision support that incorporate three stages: formulation, solution and analysis. Formulation relates to the generation of problem and domain models. Solution relates to the algorithmic solution of the model. This includes the combination of techniques from artificial intelligence and operations research to address complex problems. Analysis stage relates to the analyses and interpretation of model's solution and outcomes.

It can be observed in the IDSS models introduced above that they all incorporate a domain the knowledge component (through case bases, rule bases, knowledge acquisition subsystem, or domain models) and an intelligent system component (through an intelligent advisory system, intelligent supervisor, or model solver). Thus, it is possible to observe that some of the main features to incorporate in IDSS models are domain-knowledge and intelligent capabilities.

The concepts of intelligence and intelligent capabilities used in this research draw from the field of artificial intelligence, which argues that the intelligent behavior presented by an intelligent system relates to the abilities of gathering and incorporating domain knowledge, learning from the acquired knowledge, reasoning about such knowledge and, when required, being able to issue recommendations and justify outcomes (Schank 1982).

Thus, the required capabilities in an IDSS model can be summarized as follows:

- Incorporating specific domain knowledge
- Learning and reasoning
- Issuing recommendations
- Drawing justifications.

Domain knowledge can be classified into two types, factual and expert knowledge. Factual knowledge consists of explicit domain knowledge, such as facts, data, contexts, and relationships relevant to the decision problem; whereas expert knowledge consists of implicit domain knowledge from domain experts (Holtzman 1989).

Factual knowledge in most decision domains is complex, ill-structured and incomplete, which makes it difficult to be fully understood, formalized and incorporated into a computational system (Bonczek *et al.* 1981, Turban *et al.* 2005). On the other hand, expert knowledge acquisition from domain experts is not an easy task either. Early attempts in building expert systems revealed the difficulties in acquiring expert domain knowledge (Hayes-Roth 1994, Tecuci and Kodratoff 1995, Lenat *et al.* 1986).

A possible approach for domain-knowledge acquisition is to automatically induce specific domain knowledge directly from raw data (Fayyad *et al.* 1996, Quinlan 1993, Tecuci and Kodratoff 1995, Wu 1995). Potentially, large organizational databases contain useful information that can be used for decision-making purposes, identifying strategically important information patterns (Fayyad *et al.* 1997, Hand *et al.* 2001) Knowledge discovery in databases (KDD) is the process of extracting useful patterns and models from raw data, and making those extracted patterns understandable and suitable for the resolution of decision problems. KDD is a multistage process, in which data mining can be considered the core activity (Han, 1998), and it relates to the process and the set of techniques used to find (mine) underlying structure, information and relationships in normally large amounts of data.

Intelligent computing technologies have been applied in developing computational systems to support a wide range of problems, incorporating intelligent capabilities in these systems. For example, artificial neural networks (NN) have been explored to implement learning and reasoning mechanisms into computational systems, such as decision support systems (Wang 1994, Turban *et al.* 2005, Goonatilake and Khebbal 1995, Azvine *et al.* 2000.) NN excels in learning in uncertain or unknown environments and in performing approximate reasoning (Medsker 1995, Sun 2001).

Most of the literature about KDD relates to the development and optimization of algorithms or experiences of KDD in practice, but relatively little work has been published relating integrated approaches of KDD and intelligent computing in the context of decision support. The research described in this chapter is concerned with investigating the combination of knowledge discovery and intelligent computing technologies, in particular artificial neural networks, in developing a framework for decision support.

From that perspective, this research has concentrated on investigating how data mining and neural networks can cooperate in order to minimize problems related to knowledge acquisition, reasoning, and learning in building decision support systems. As a result of this investigation a model for intelligent decision support system is proposed combining an association rule-generator algorithm for data mining with an artificial neural network based system in a hybrid architecture (Viademonte 2004) Within this architecture, data mining is applied to induce expert-domain knowledge from organizational databases, minimizing the problem of knowledge acquisition. The discovered association rules are stored in a rule-based knowledge base. A neural network-based system is introduced to provide learning and problem solving, taking advantage of the neural network capability of generalization, handling large combinations of data, and coping with noise data.

To assess the performance of the proposed framework for IDSS, it has been implemented in the context of aviation weather forecasting, identifying severe and rare weather phenomena at airport terminals, particularly fog phenomena. Refer to (Viademonte *et al.* 2001a) for discussion on knowledge discovery in aviation weather forecasting.

The results achieved demonstrated that the proposed IDSS model constitutes suitable technology to implement and deploy intelligent decision support systems.

This chapter is organized in 4 sections, as follows: section 4.1 introduces the research described in this chapter and gives some theoretical background. Section 4.2 introduces the proposed IDSS framework, its components, their respective roles and interactions, including the employed neural network environment and neural network model. Section 4.3 describes the knowledge representation schema, addressing issues of domain modeling in aviation weather forecasting. Section 4.4 discusses issues about the functionality of the proposed IDSS framework, and Section 4.5 presents the conclusions and directions for further research.

4.2 A Framework for IDSS

The proposed framework for IDSS combines data mining (DM) and artificial neural network (ANN) modules in a hybrid architecture called the DM-NN model, and it has been developed and applied to an industry problem to empirically assess its applicability (Viademonte and Burstein, 2001, Viademonte 2004)

The aim of the proposed framework for IDSS is to support decision making by recalling past information, inducing "chunks" of domain knowledge from this information and performing reasoning upon this knowledge in order to reach conclusions in a given classificatory situation. The proposed model for IDSS has to be capable of building domain knowledge from data-rich domains and applying this knowledge in problem solving. It was designed as a predictive tool for classification problems. It aims to predict which class a given case falls within. Examples of such situations are medical diagnoses, where the objective is to diagnose a particular disease based on a set of observed symptoms. Or weather forecasting, where the likelihood of the occurrence of a particular weather phenomenon is determined based on a set of weather observations.

There are two main stages in the operation of the DM-NN model. First, descriptive models about the application domain are built. Next, predictive models of this domain are built. Figure 4.1 illustrates these stages:

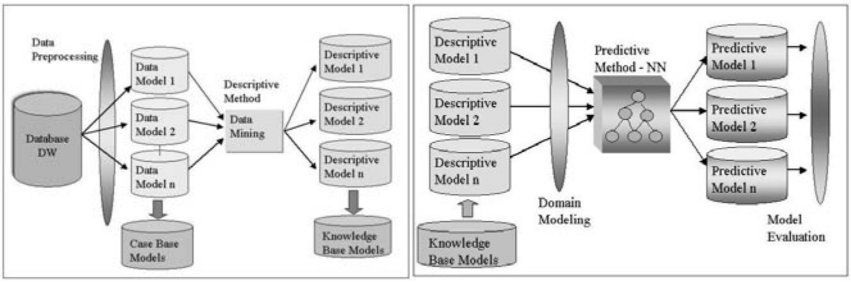


Figure 4.1. Building descriptive and predictive models

In the DM-NN model operation raw data are extracted from databases, pre-processed, and sets of cases are obtained as a result of this process; these sets of cases are named data models. Data models are used as input data into a particular *descriptive method* (a data-mining algorithm) in order to build descriptive models. Descriptive models are stored in knowledge bases. Next, the descriptive models are used as input data into a *predictive method* (a neural network model). The results of the neural network processing (predictive method) are the predictive models.

4.2.1 The DM-NN Model Architecture

The DM-NN model has a multilayered architecture that can be divided into two levels: *data* and *process*. At the process level it applies data mining for knowledge acquisition and a neural network-based system as a core for an advisory system.

Specifically, data-mining technology was chosen to induce expert-domain knowledge from historical databases, hence minimizing the difficulties of acquiring expert domain knowledge, and a ANN based system is employed to implement learning and reasoning with the knowledge obtained through data mining. The employed ANN system also provides explanatory capabilities, and the user-interface level.

At the data level the DM-NN model comprises all the data repositories used during the various stages of decision support; it includes a decision-oriented data repository (ideally a data warehouse), case bases and knowledge (rule) bases. The basic computational elements of the DM-NN architecture are:

- A decision-oriented data repository, such as a data warehouse
- Case bases
- Inductive algorithm for data mining (descriptive method)
- Knowledge bases
- An intelligent advisory system (predictive method)

Figure 4.2 illustrates the main components of the proposed architecture and the way they interact.

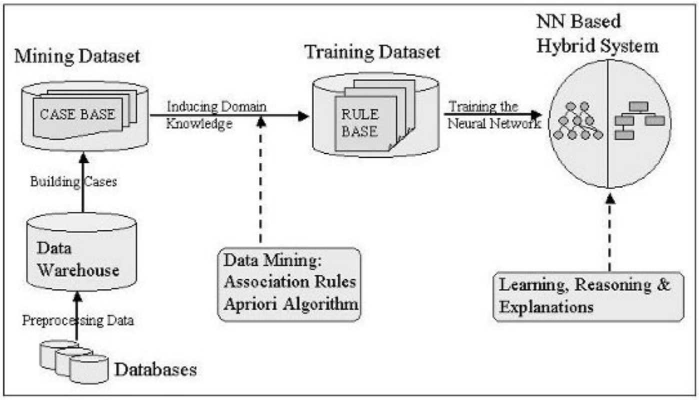


Figure 4.2. The components of the DM-NN architecture as in (Viademonte, 2004)

The dotted lines in Figure 4.2 represent processes among the components; for instance, the dotted line from the data-mining box indicates that a data-mining process happens in order to induce domain knowledge from case bases. The second dotted line indicates the capabilities implemented by the ANN-based system: learning, reasoning and explanation. The dashed lines in Figure 4.2 represent data flows between the components. For example, historical raw data from databases are preprocessed and fed into the data warehouse. From there, cases are selected and extracted, and then stored in case bases.

A decision-oriented data repository is introduced in the DM-NN architecture as the primary source of information, and ideally it should be a data warehouse. Data-warehousing technology is introduced in the DM-NN model to overcome the problems related with transactional data used in high level decision support tasks, *i. e.* to transform transaction-oriented data to decision support-oriented data.

Case bases contain selected instances of relevant cases from the specific application domain; they consist of preprocessed sets of raw (historical) data used as input in the data-mining component. As such, case bases are also termed *data models* (or *mining datasets*). A case consists of a set of feature/value pairs and a class in which the case falls.

The case base is a fundamental component in the proposed DM-NN model. The ability to build relevant cases can lead to the success or failure of a particular application. For that reason it is proposed that cases should be built from the data stored in a data warehouse, as this can ensure consistency of data.

The DM-NN model was applied in aviation weather forecasting; as such, in this research, cases are series of weather observations.

Knowledge-rule bases are built based on data-mining results; they contain structured generalized knowledge that corresponds to relevant patterns (associations) found (mined) in the case bases. The knowledge obtained as a result of data-mining trials is termed *knowledge models* (*training datasets*), and is stored in knowledge rule bases. In the context of the DM-NN model, knowledge bases are accessed by neural networks for learning purposes, and as such they constitute the training datasets.

A data-mining component is introduced to implement knowledge acquisition through cases stored in case bases, and a hybrid (symbolic-connectionist) system is applied to process the obtained knowledge, implementing learning, reasoning and explanatory capabilities. These components are discussed next.

4.2.2 Data-mining Component

The DM-NN model applies data mining to discover relevant relations out of the case bases. In this approach *specific knowledge* is represented in the form of *cases*, from where *general knowledge* is derived in the form of *association rules*.

Sets of cases are presented to a data-mining component to discover "chunks" of knowledge about a particular domain. This combination of data mining and case bases is suggested to implement knowledge acquisition in the proposed decision support model (Viademonte and Burstein 2001), handling one of the bottlenecks in developing intelligent systems, the knowledge acquisition from human experts. The idea of using cases to perform knowledge acquisition is based on the assumption that a conceptualized part of knowledge about a certain domain is represented as cases (Kolodner 1993). Consequently, it is possible to induce relevant pieces of knowledge (chunks) from a certain domain from sets of cases about that domain.

In the DM-NN model the relations obtained from the cases are represented as association rules and stored in knowledge rule bases. An association rules generator algorithm is employed for data-mining purposes. This is an implementation of the Apriori algorithm for association rules (Agrawal *et al.* 1993.) Briefly, an association rule is an expression X→Y, where X and Y are sets of predicates; X being the precondition of the rule in disjunctive normal form and Y the target-post condition. Association rules have two attributes, a confidence measure and a support measure. The rule confidence is the conditional probability with which predicates in Y are satisfied by a tuple (record) in the database given that predicates in X are satisfied. Such a rule is said to be frequent if its frequency exceeds a predefined threshold, *e. g.* if all predicates $X \cup Y$ occur together at least a user-specified minimum number of times (Agrawal *et al.* 1993).

Figure 4.3 illustrates the proposed knowledge acquisition process through data mining.

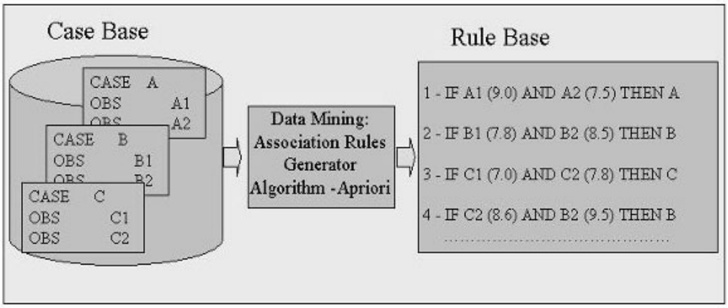


Figure 4.3. Knowledge acquisition through data mining

4.2.3 Intelligent Advisory System Component

It is the purpose of this research to implement in the IDSS framework the capabilities previously discussed in section 4.1, *i. e.* incorporating specific domain knowledge, learning and reasoning, issuing recommendations and drawing justifications.

The intelligent advisory system (AIS) component is responsible for the implementation of these capabilities. It is said to be advisory as it offers suggested choices to the user decision maker together with respective justifications. The specific architecture of this advisory system is hybrid (symbolic-connectionist) (Sun 2001, Medsker 1995), as it combines ANN models within a symbolic mechanism for knowledge representation, according to what is discussed in Section 4.3, “Representing Knowledge in the DM-NN Model.”

The IAS component is capable of learning from data, and reasoning about what was learned through its neural network mechanism. And it is able to justify its reasoning through its symbolic knowledge representation mechanism, which cooperates with the ANN model.

The IAS component uses knowledge stored in knowledge bases to learn about a particular problem, this is why this research treats knowledge bases (see previous section) as training datasets. After the neural-network training process has been completed the system is capable of reasoning about the problem within the boundaries of the knowledge it obtained, and it is ready to be used as an advisory decision support system. Figure 4.4 shows an overview of the internal architecture of such a system, its main components and processes.

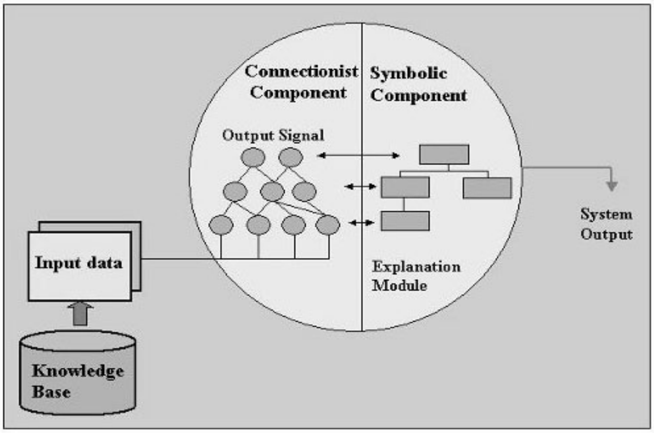


Figure 4.4. Internal architecture of the IAS component as in (Viademonte, 2004)

The IAS component implements learning and reasoning through the CNM neural network model (Machado and Rocha 1990), and explanatory capabilities through a framework architecture for decision support systems, which combines ANN models with a symbolic mechanism for knowledge representation (Beckenkamp 2002), as a result of this symbolic knowledge representation the IAS is able to draw justifications about its output.

4.2.3.1 The Combinatorial Neural Model

The Combinatorial Neural Model (CNM) (Machado and Rocha 1990, 1992) was developed and explored during the last decade. It was inspired by the knowledge acquisition methodology of knowledge graphs (Leao and Rocha 1990), which was developed to provide means for the representation and combination of knowledge elicited from multiple experts.

The CNM is an acyclic multilayer feedforward network. It is usually implemented with three layers: an input layer, a hidden layer, and an output layer. The output layer contains neurons that represent different hypotheses (classes) involved in a particular problem; the input layer contains neurons that represent the domain information that supports the output classes, and the hidden layers specify different combinations of input neurons than can lead to a particular class.

Input neurons are formed by fuzzy values in the interval [0,1], indicating the degree of confidence (or measure of relevance) of the information represented by each input neuron. Neurons are linked by connections, *e.g.*, synapses. CNM implements two types of synapses: excitatory and inhibitory. Excitatory synapses propagate an input signal using their synaptic weight X as an attenuating factor. Inhibitory synapses implement a fuzzy negation on the arriving signal, transforming it into 1-X. Then signals are propagated by multiplying its value by the synaptic weight. Combinatorial neurons propagate incoming values according to a fuzzy AND operation.

CNM implements a supervised learning approach based on the error correction algorithm, similar to the backpropagation, in which punishment and reward accumulators are computed for each connection in the network and the current connection weights are computed through the normalization of those accumulators. During the learning phase (training), as each example is presented and propagated, all links that led to the right classification have their reward accumulator incremented; otherwise, misclassifications increment the punishment accumulators. At the end of the learning phase, connections with higher punishment values than reward values are pruned. The remaining connections have their weights updated using the accumulators.

Once the CNM is trained, it pursues the following strategy to come up with a decision for a specific case. The CNM evaluates the given case and calculates a confidence value for each hypothesis. The inference mechanism finds the winning hypothesis, the one with the highest confidence value, and returns the corresponding result.

Detailed discussion about the CNM model and its learning algorithms can be found in (Machado and Rocha 1990, Machado *et al.* 1998) The CNM has been successfully employed in several experiments dealing with classification problems, and these experiments are reported by (Leao and Reategui 1993, Reategui and Campbell 1995, Viademonte *et al.* 1995)

4.2.3.2 Components for Artificial Neural Networks

The IAS component has been implemented through the components for artificial neural nNetworks (CANN) framework (Pree *et al.* 1997, Beckenkamp 2002) CANN implements a framework architecture for decision support systems, which design combines ANN models with a symbolic mechanism for knowledge representation. As a result of this symbolic knowledge representation, the IAS is able to draw justifications about its output. Once a particular output neuron is fired, the IAS recovers the input neurons and the pathway that led to the result, identifying explicitly the information content of those neurons.

The CANN components are object-oriented designed; hence the core parts are done as small frameworks. The Neural Network framework is defined in order to facilitate the implementation of different ANN models. This is achieved by modeling the core entities of the ANN (neurons and synapses) as objects and storing the generated ANN topologies as objects via Java's serialization mechanism. As such, CANN is able to reuse these core ANN components for implementing new ANN models. Detailed discussion about the ANN framework can be found in (Beckenkamp 2002)

Particularly relevant to this research is the domain framework. In CANN, the domain is represented through four main classes: *domain*, *evidence*, *hypothesis*, and *attribute*. Figure 4.5 illustrates the class hierarchy for domain representation.

Evidences form the input data, and experts use evidences to analyze the problem in order to arrive at decisions. Evidences in aviation weather forecasting would be the *levels of rainfall*, *wind speed* and *direction*, for example. *Evidences* are described by their respective *attributes*, so the *attribute* class was incorporated in the framework. One or more *attribute* objects describe the value of each *evidence* object.

For example, in the case of the evidence *wind direction*, this might be defined as a set of string values (string attributes) such as North, Northwest, South, Southwest, *et al.* The *attribute* class is subclassed according to the different data types an attribute might hold, such as numeric, string and fuzzy sets (see Figure 4.5.)

Furthermore, the classification categories (or *hypotheses*) constitute a further core entity of such problems. In aviation weather forecasting, hypotheses would be fog occurrence, thunderstorms, cyclones, *et al.* The *hypothesis* class represents the possible classes (or hypotheses) involved in a particular application.

In CANN, an instance of class *domain* represents the problem by managing the corresponding *evidence* and *hypothesis* objects. Figure 4.5 shows the class hierarchy involved in the problem-domain representation.

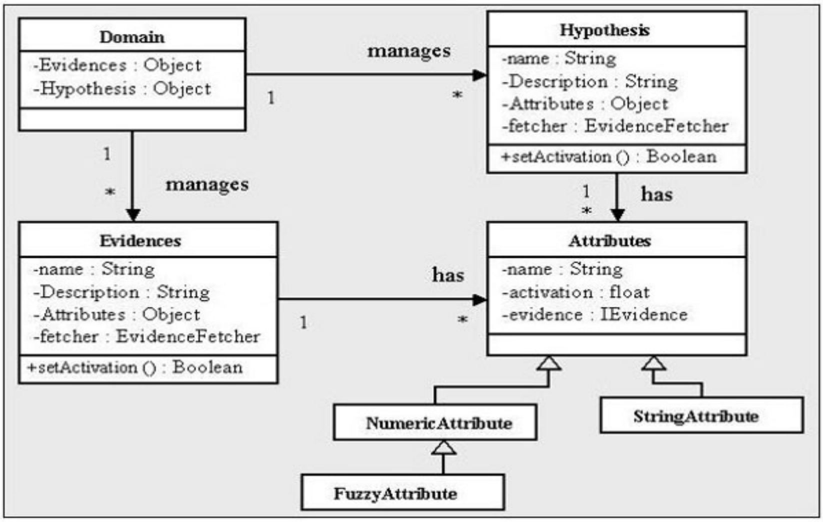


Figure 4.5. Class hierarchy for domain representation, as in (Beckenkamp, 2002)

Besides the neural network and domain frameworks, CANN defines a framework for processing problem-specific data. Fetcher and EvidenceFetcher abstract classes constitute the framework for processing problem-specific data. Readers interested in a more comprehensive and detailed discussion about the CANN project and framework should refer to (Beckenkamp 2002; Pree *et al.* 1997)

4.3 Representing Knowledge in the DM-NN Model

The previous section introduced the various components employed in the DM-NN model. As important as the technological components is the way domain knowledge is represented within those components. This section discusses the knowledge representation mechanisms used in the DM-NN model, illustrating this representation through the aviation weather forecasting domain.

Firstly, it is important to mention that the knowledge-representation schema employed in the DM-NN model is grounded on the knowledge-acquisition methodology of knowledge graphs (Leao and Rocha 1990) A knowledge graph (KG) is defined as a directed AND/OR acyclic graph used to represent expert knowledge for a particular classification hypothesis. There are three types of nodes in a KG: *hypothesis nodes* represent the hypotheses, or classes, considered in the graph; *evidence nodes* represent input information that supports a particular hypothesis, and *intermediate nodes* represent different groupings of evidences that lead to a specific hypothesis or class. These groups of evidence represent chunks of knowledge applied by an expert when reasoning about a problem. Intermediate nodes represent a logical AND operation among the evidence nodes linked to them (see Figure 4.6)

The KG structure is very similar to the CNM topology, in which hypothesis nodes can be mapped into output neurons, evidence nodes into input neurons and intermediate nodes mapped into combinatorial neurons (hidden layer)

The domain knowledge is represented in three ways in the DM-NN model:

- Through association rules
- Through a neural network model, *i. e.* implicit in the neural network structure
- Through a hierarchy of classes and objects.

4.3.1 Representing Knowledge Through Association Rules

Domain knowledge is presented through association rules at the data mining level. Figure 4.6 shows part of a KG from the weather-forecasting domain:

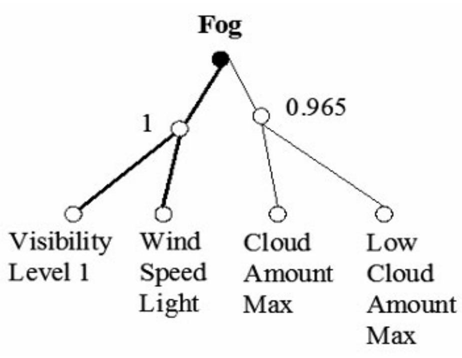


Figure 4.6. Knowledge graph from the aviation weather forecasting domain

A KG can be translated into a rule representation, associating the respective evidence nodes. For example, the KG from Figure 4.6 can be represented as follows:

If *(Visibility is level 1)* *AND (Wind Speed is Light)*

THEN *(fog is predicted with 100% degree of confidence)*

This notation is very similar to the notation of association rules (refer to Section 4.2.2 for a description of association rules). As such, the association rule representation of the first pathway of Figure 4.6 can be described as:

$X = \{E1 , E2\}$, *and* $Y = \{ FOG \}$, *rule confidence*= *100%*
Where $X \rightarrow Y$
$E1$ = *visibility level 1*
$E2$ = *wind speed light*

Consequently, knowledge graphs can be represented by sets of association rules, similar to the ones generated by the Apriori algorithm (Agrawal *et al.* 1993.) Furthermore, association rules can be automatically induced from cases, through an association rule generator algorithm. This approach might represent a potential solution to the problem of knowledge acquisition and representation through knowledge graphs. Although KGs constitute a powerful approach for knowledge acquisition and representation in classification problems, its construction is time consuming and involves a costly process, requiring the assistance of at least one domain expert (Viademonte *et al.* 1995)

Therefore, association rules were chosen as the knowledge representation formalism in the DM-NN model for their similarity with knowledge graphs, because they represent a clear and natural way of knowledge representation that is easy for people to understand, because they easily represent simple causalities that are suitable for the meteorological domain, because there are efficient algorithms for association rules discovery, and finally because they fit smoothly into the selected neural network model, the CNM.

4.3.2 Representing Knowledge Through Neural Networks

Neural networks, particularly the CNM model, were selected to implement learning and reasoning capabilities in this research, as ANN are good at implementing lower-level reasoning. They excel in recognizing complex patterns, learning and generalization from examples and have powerful self-organizing capabilities.

The CNM was selected because of its compatibility with KGs and, as a result, association rules. Additionally, the CNM has been successfully employed in several experiments dealing with classification problems, such medical diagnoses (Leao and Reategui 1993), credit card scoring (Reategui and Campbell 1995) and engineering problems (Viademonte *et al.* 1995).

Regarding its similarity with the KGs, association rules can also be mapped into the CNM topology. In this case, each evidence/attribute value pair corresponding to a rule's antecedent items is mapped onto an input neuron in the CNM topology. The right side of the rule, *i. e.* the consequent item, is mapped on to an output neuron; and the rules correspond to the strengthened connections among the input nodes, *i. e.* the CNM combinatorial (hidden) layer. For instance, rules describing relations in the weather forecasting domain are represented by neurons and synapses. Figure 4.7 illustrates this property.

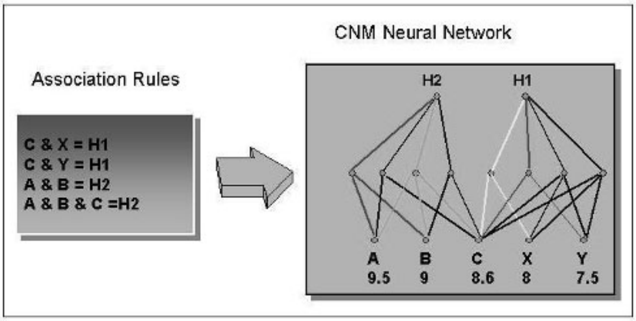


Figure 4.7. Mapping rules into the CNM topology

In Figure 4.7, the antecedent items of the rules, the evidences A, B, C, X and Y, are represented as input neurons in the CNM structure, and the hypotheses H1 and H2 are represented as output neurons. For instance the rule antecedent item A is mapped on the first input neuron, item B is mapped on the second input neuron, and item C is mapped on the third input neuron. Additionally, importance degree values can be assigned to each rule antecedent item. These importance degree values can be then transferred as input neuron weights in the CNM structure. Synaptic weights are calculated by the CNM algorithms, and hidden neurons correspond to combinations of evidences, representing rules. The leftmost hidden neuron in Figure 4.7 represents the rule If A and B than H2, indicating to hypothesis H2, represented by the output neuron with the same name.

4.3.3 Representing Knowledge Through a Hierarchy of Classes and Objects

According to what was previously discussed in Section 4.2.3.2, in neural network models knowledge is implicitly represented as connection weights distributed across the ANN topology. In such a knowledge representation schema it is very difficult to explicitly access that knowledge for explanatory purposes. To minimize this problem CANN implements an architecture in which ANN structures, including the knowledge stored across these structures, are symbolically represented in a hierarchical fashion, through an object-oriented design.

The domain of aviation weather forecasting was modelled through a hierarchy of classes, describing the domain classes, evidences, attributes and the interrelationships among them, according to the approach illustrated in Figure 4.5. Then, this hierarchy of classes is integrated into CANN. Once the domain is modeled, CANN accesses the respective knowledge bases (refer to Section 4.2.1) for learning purposes.

Figure 4.8 illustrates how the aviation weather forecasting domain was modeled according to this approach. The main classes are *domain*, *evidences*, *attributes* and *classes*. Specifically, *hypothesis* are presented by instances of *classes*, *evidences* are represented by instances of *evidences* and *attributes*, and intermediate nodes (according to the KG structure) are represented by aggregation between instances of *evidences/attributes* and *classes*.

In Figure 4.8 *domain* is the higher level class, from where different application domains can be subclassed, through "is-a" relationships. In this case, the aviation weather forecasting domain is created as an instance of *domain*. The *domain* class is connected by "part of" arcs to *evidences* and *classes*, implying that the *domain* should be made by the composition of these two classes. Therefore, the aviation weather forecasting domain is made by the composition of instances of *classes* and *evidences*.

Instances of *classes* (it is important to make the distinction between the object-oriented definition of class, and the definition of *classes* as part of a classification problem) in the aviation weather forecasting domain are weather phenomena such as fog, thunderstorms, and cyclones.

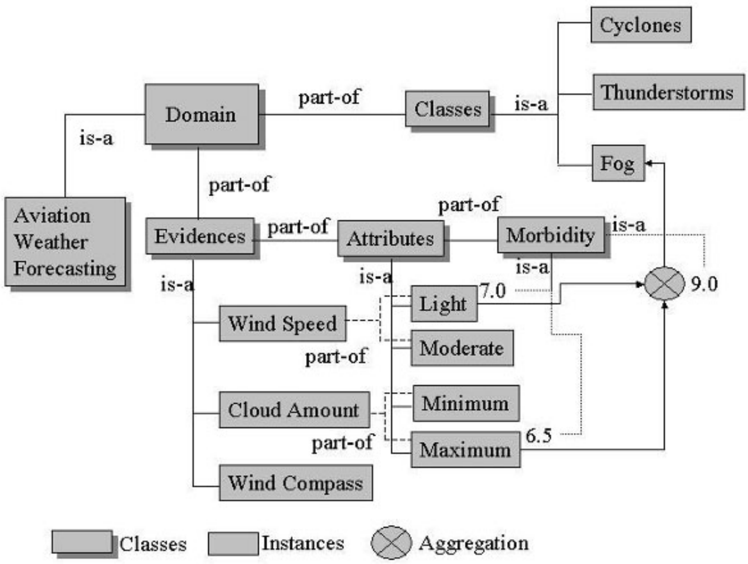


Figure 4.8. The aviation weather forecasting domain model as in (Viademonte 2004)

Following Figure 4.8, *evidences* (class) is connected to *attributes* by a "part of" relationship, implying that *evidences* instances are composed of instances of *attributes*. Instances of *evidences* class are *Wind Speed*, *Cloud Amount* and *Wind Compass*, connected to the *evidences* class by an "is-a" arc. Similarly, instances of *attributes* class are *Light* and *Moderate* that are part of the *Wind Speed evidence*. In the same way, *attribute* instances *Minimum* and *Maximum* are part of the *Cloud Amount evidence*.

Associations between *evidences/attributes* that together lead to instances of *classes* are represented by the concept of aggregation. For example, the aggregation of the *evidence/attribute* pairs *Wind Speed Light* and *Cloud Amount Maximum* leads to fog.

Pairs of *evidences/attributes* have a morbidity value assigned to them. Morbidity is a value that indicates the importance of a particular pair evidence/attribute related to instances of *classes*. Similarly, aggregations have a morbidity assigned to them, which indicates the importance of a particular *evidence/attribute* association related to the class indicated by that association.

In Figure 4.8, the value of 7.0 (in a scale from 0 to 10) assigned to the *evidence/attribute* pair of *Wind Speed Light* means that this *evidence/attribute* pair has a significant importance in classifying fog cases. In a similar fashion, the value of 9.0 assigned to the association between *Wind Speed Light* and *Cloud Amount Maximum* indicates that this association strongly contributes to a positive case of fog.

The morbidity is modeled as a class connected to the *attributes* class through "part of" arcs. The concept of morbidity to measure the importance of *evidences* in classification problems used in this research is based in the morbidity scale used by the Internist-I system (Miller 1986) and later by the hybrid case-base reasoning model developed by Reategui (1997)

Through this modeling schema the domain of aviation weather forecasting can be integrated into the domain framework implemented in CANN.

4.4 The Functionality of the DM-NN Model

The functionality of the DM-NN model can be seen from two perspectives: as an iterative and interactive decision support *process* and a *computational architecture*. Firstly, it defines a decision process, and at the same time it provides a computational architecture for linking various technological components in a single decision support cycle (see Figure 4.9)

From the process perspective it proposes a line of actions that can be taken to support a particular decision situation. In that sense it is a normative process, as one activity relies on the previous activity linked by some algorithmic relationship. Despite this normative aspect, it is not necessary for the decision maker to follow all the proposed steps until the final recommendation is reached. If a decision maker is satisfied with intermediary results, the process can be stopped at that level.

At the same time, the proposed model for decision support involves a computational architecture, as it suggests the combined use of different computational components and technologies. As a result, it defines an interactive computational environment that uses data-mining technology to automatically induce domain knowledge from case bases, and an ANN-based system as a core for an advisory system, which provides the user interface.

The system provides three levels of decision support: *rules generation*, *case consult* and *case-base consult*. The rule generation corresponds to the set of association rules generated in a data-mining session, which may be evaluated by the decision maker. If the generated set of association rules provides enough information to the user decision maker to arrive at a decision, the situation is resolved and the generated rules can be stored in the knowledge base for further use.

At this point the process can be considered finished. Otherwise, the rules can be presented to the ANN for learning.

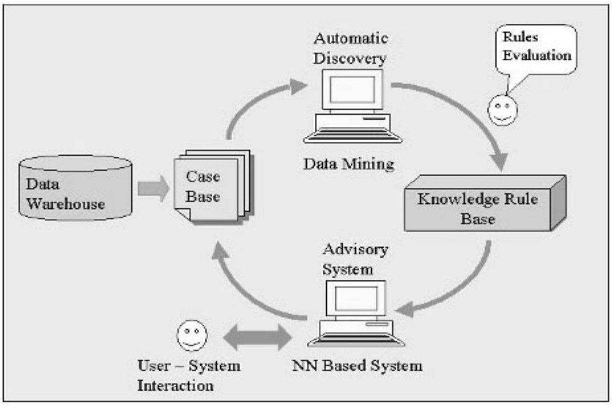


Figure 4.9. The decision support cycle of the DM-NN model

After the NN-based learning procedure has been executed, the advisory component provides a case consult and a case-base consult facility through its consult mode for the user to test and validate hypotheses about the current decision situation.

A case consult presents a selection of *evidences* and their respective evaluation by the IAS component. The IAS component evaluates the selected evidences and calculates a confidence degree for each *hypothesis*. The inference mechanism implemented by the CNM neural model indicates the *hypothesis* with the highest confidence degree as the candidate solution (or class) to the problem.

Figure 4.10 illustrates a case consult in aviation weather forecasting, where four *evidences* are selected from the list of evidence: *Dry Bulb Low*, *Cloud Amount Max*, *Sea Level Pressure Vhigh (very high)* and *Wind Speed Light*. The list in the left side of the window shows all the *evidences* modeled for the domain being analyzed. The list in the right side of the window shows the evidences that were selected for evaluation. When clicking the *Test Case* button the ANN is activated and evaluates the selected *evidences*.

The bottom part of the window in Figure 4.10 shows the ANN output to the presented case. In this example the ANN evaluated fog *hypothesis* as the correct class (winner hypothesis). The explanation for this is the simultaneous occurrence of *evidences Cloud Amount Max* and *Wind Speed Light*, with a computed confidence degree of 0.953. This means that, based on what was learned from the training sets, these evidences strongly contribute to a fog occurrence. And strongly is here quantified as 0.953, based on the ANN-reasoning algorithm (Machado *et al.* 1998)

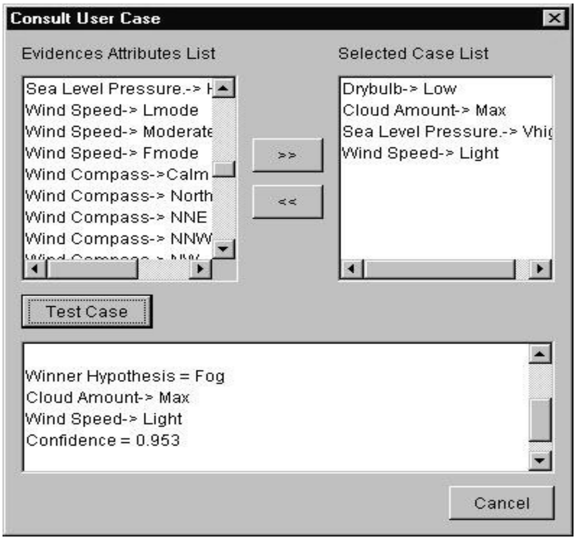


Figure 4.10. A fog case consult into CANN

A case-base consult is similar to a case consult, except that instead of presenting a single case (or one set of evidences) each time, several cases are presented to the IAS component for evaluation. The IAS component evaluates a set of cases in the same way as a single case.

Furthermore, if the user decision maker believes that the outcome of the IAS (after a consult has been performed) represents novel and potentially useful information; it can be stored, either in the case base as a new case, or in the knowledge base as a new rule.

4.5 Evaluation and Conclusions

The IDSS framework proposed in this research investigated the combination of knowledge discovery in database and intelligent computing technologies. In this context, the activities related to knowledge discovery in databases, such as data preprocessing, selection, cleaning and data sampling. Also, the activities related to data mining such as features selection, numerical data discretization and setting data mining parameters such as the level of rule confidence and rule support degrees need to be considered. The discussion of these activities is not in the scope of this chapter, but they certainly need to be taken into account. For detailed discussion on subjects of data pre-processing and data preparation for data mining in aviation weather forecasting, refer to (Viademonte *et al.* 2001, Viademonte 2004).

The performance of the DM-NN model for IDSS has been assessed according to its capability of correctly classifying meteorological observations, specifically fog cases. It is a quantitative approach where the holdout method (Weiss and Indurkhya 1998) was employed.

As part of the data-sampling procedures and the selection of data-mining parameters, various training and testing data sets were obtained. For the purpose of illustratating the achieved results, part of the obtained training sets is presented here. Table 4.1 lists six training data sets, their classificatory rates, the number of cases for which a conclusion could not be reached, and the error rate in the testing set.

Table 4.1. Performance on fog classification

Training set	Not evaluated	Classificatory rate	Error rate	Rate on fog classification
TrainSet1	1.88%	76.88%	0.23	81.67%
TrainSet2	2.50%	76.88%	0.23	81.67%
TrainSet3	2.50%	78.13%	0.22	68.33%
TrainSet4	1.88%	78.75%	0.21	86.67%
TrainSet5	1.88%	76.88%	0.23	83.33%
TrainSet6	2.50%	76.25%	0.24	76.67%
AVERAGES	2.19%	77.29%	0.23	79.72%

It can be observed in Table 4.1 that the average performance in fog classification was 79.72%, with the best result achieved when using the TrainSet4 as training set, with 86.67% of correct fog cases classified. It is important to recall that the DM-NN computational model is proposed as an iterative and interactive environment for decision support (refer to Figure 4.9).

The application of such an approach requires a series of activities in its diverse stages, for example, gathering information about a particular decision problem, analyzing such information and preparing data, as well as choosing an adequate technology for mining data, evaluating outcomes and populating knowledge bases. The necessity of gathering new data or making changes in the domain are also considered (even expected), as discovered knowledge is likely to give new insights about new information to be collected or better ways to model the problem under study.

The IAS component is likely to require a series of interactions until it achieves its best performance or a stable level of performance, as problem situations are dynamic. Consequently, problem models are expected to change and adapt over time. Domain modeling also has to be taken into account. Different ways of modeling the problem might result in better performances than the ones achieved in this research.

The results obtained can be considered satisfactory for fog identification. According to a study developed by (Keith 1991) forecasts for Tullamarine airport (Melbourne, Australia) demonstrate poor performance for low stratus and fog. This study considers 5-year means for various airport cities in Australia, taking the latest 5-month running means of the probability of detection (POD) and false alarm ratio (FAR) for low cloud cases including fog. Tullamarine showed the worst POD with 69% and a FAR of 77%.

The experiments illustrated in Table 4.1 resulted in fog classificatory performance average of 79.72%, with the best individual performance of 86.67% when applying TrainSet4. Figure 4.11 contrasts these results.

These results are indicative of a higher performance achieved by the DM-NN approach when contrasted with the results reported by Keith's study (Keith 1991) However, the main aim of the experiments and results conducted in aviation weather forecasting is to provide means for assessing the feasibility and applicability of the DM-ANN approach for decision support, rather than to come up with optimal results.

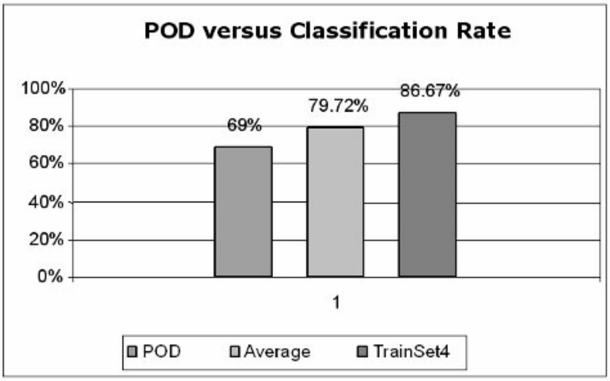


Figure 4.11. Contrasting the POD with classificatory rates

The research described in this chapter aims at achieving satisfactory results, where satisfactory results are defined by the user decision makers based on their own utility functions about the novelty or usefulness of the outputs given by the proposed approach for decision support. The individual performance of the data-mining and neural network algorithms are not the main concerns in this research. This research is concerned with the combined approach, with the DM-NN IDSS model performance as a whole, its usefulness, suitability and effectiveness in a decision-making problem, rather than the performance of a single technology by itself.

Consequently, the results obtained through the classificatory performance provide directions regarding the DM-NN model's applicability as a decision support framework for a data-rich domain, more importantly they demonstrated that a combination of data mining through an association rule generator algorithm and artificial neural networks was capable of providing a model for decision support systems that automatically builds domain knowledge from organizational databases, and performs learning and reasoning upon that knowledge in supporting decision-making.

Some subjects for future work were identified. For example, to investigate the possibilities of expanding the DM-NN architecture towards a more integrated approach. One of the possibilities in that direction would be to coordinate the interactions between the diverse components of the architecture. This can be achieved by integrating a manager component in the DM-NN architecture, to coordinate the operation between the data-mining and neural network components.

Another possibility is to investigate the extension of the CANN framework, implementing other reasoning mechanisms besides the connectionist one.

Another interesting research subject would be the definition of structures to represent domain modeling and knowledge through XML descriptions. This would facilitate an integrated architecture for the DM-NN model, bringing a higher level of flexibility into the architecture. Additionally, expanding the DM-NN architecture also brings the possibility of investigating alternative reasoning approaches besides the connectionist.

References

Agrawal R, Imielinsk T, Swami A (1993) Mining association rules between sets of items in large databases. ACM SIGMOD Conference on Management of Data (SIGMOD'93), (pp. 207-216). New York, NY: ACM Press.

Azvine B, Azarmi N, Nauck D (2000) Intelligent Systems and Soft Computing: Prospects, Tools and Applications. (Eds. Carbonell, J. and Siekmann, J.), Lecture Notes in Artificial Intelligence. Adastral Park, UK: Springer-Verlag.

Beckenkamp FG (2002) A Component Architecture for Artificial Neural Network Systems. Faculty of Sciences, Department of Computer and Information Science, University of Constance, Germany. Ph.D. Thesis.

Bonczek RH, Holsapple CW, Whinston AB (1981) Foundations of Decision Support Systems. New York: Academic Press.

Burstein F, Smith H, Sowunmi A, Sharma R (1998) Organisational Memory Information Systems: a Case-based Approach to Decision Support. In (Eds, Kersten, G., Mikolajuk, Z., Rais, M. and Yeh, A.), Decision Analysis and Support for Sustainable Development, (Chapter 20).

Fayyad UM, Piatetsky-Shapiro G, Smyth P, Uthurusamy R (1996) From data mining to knowledge discovery: an overview. In Advances in Data Mining and Knowledge Discovery, (pp. 1-34) Cambridge: AAAI/The MIT Press.

Fayyad UM, Mannila H, Ramakrishman R (1997) Data Mining and Knowledge Discovery. Boston: Kluwer Academic Publishers.

Goonatilake S, Khebbal S (1995) Intelligent hybrid systems: issues, classifications and future directions. In (Eds, Goonatilake, S. and Khebbal, S.), Intelligent Hybrid Systems, (pp. 3-20). New York NY: John Wiley & Sons.

Han J (1998) Data Mining: An Overview from Databases Perspective. Tutorial on the Pacific-Asia Conference in Knowledge Discovery and Data Mining (PKDD-98), Melbourne, Australia. April 1998.

Hand D, Mannila H, Smyth P (2001) Principles of data mining. (Ed, Dietterich, T.), Adaptive Computation and Machine Learning Series. Cambridge: The MIT Press.

Hayes-Roth, F. and Jacobstein, N. (1994). The state of knowledge-based systems. Communications of the ACM, 37 (3); 27-39.

Holtzman S (1989) Intelligent Decision Systems. (Eds, Buchanan, B., Davis, R., Erman, L.D., King, D., McDermott, J. and Stefik, M.), The Teknowledge Series in Knowledge Engineering. Menlo Park California: Addison-Wesley.

Keith R (1991) Results and Recommendations Arising From An Investigation Into Forecasting Problems At Melbourne Airport. Townsville, Australia, Bureau of Meteorology, March 1991. Meteorological Note 195.

Kolodner J (1993) Case-based Reasoning. Mountain View, CA: Morgan Kaufmann.
Leao BF, Rocha A (1990) Proposed methodology for knowledge acquisition: a study on congenital heart disease diagnosis. Methods of Information in Medicine, 29: 30-40.
Leao BF, Reategui EB (1993) A hybrid connectionist expert system to solve classificational problems. Proceedings of Computers in Cardiology. London: IEEE Computer Society.
Lenat DB, Prakash M, Shepherd M (1986). Cyc: using common sense knowledge to overcome brittleness and knowledge-acquisition bottlenecks. AI Magazine, Winter 1986, 6: 65-85.
Machado RJ, Rocha AF (1990) The combinatorial neural network: a connectionist model for knowledge based systems. In (Ed, Bouchon-Meunier, B., Yager, R.R. & Zadeh, L.A.), Uncertainty in knowledge bases. Berlin, Germany: Springer Verlag.
Machado RJ, Rocha AF (1992). A hybrid architecture for fuzzy connectionist expert systems. In (Ed, Kandel, A. & Langholz, G.), Hybrid Architectures for Intelligent Systems (pp. 135-152). Boca Raton FL: CRC Press.
Machado RJ, Barbosa C, Neves A (1998) Learning in the combinatorial neural model. IEEE Transactions on Neural Networks, 9 (5): 831-847
Medsker LR (1995) Hybrid Intelligent Systems. Kluwer Academic Publishers.
Miller RA (1986) Internist-I: an experimental computer-based diagnostic consultant for general internal medicine. In (Eds, Reggia, J.A. and Stanley, T.), Computer-assisted medical decision-making, 2, 139-158. New York: Springer Verlag.
Pree W, Beckenkamp FG, Rosa S.I.V. (1997) Object-oriented design & implementation of a flexible software architecture for decision support systems. Proceedings of the 9th. International Conference on Software Engineering & Knowledge Engineering (SEKE'97), (pp. 382-388). Madrid, Spain. June 1997.
Quinlan JR (1993) C4.5: Programs for Machine Learning. San Mateo, CA.: Morgan Kaufmann.
Reategui EB, Campbell J (1995) A classification system for credit card transactions. Advances in Case-Based Reasoning: Second European Workshop (EWCBR-94), (pp. 280-291). Chantilly, France. Berlin, Germany: Springer Verlag.
Reategui EB (1997) Combining Case-Based Reasoning with Neural Networks in Diagnostic Systems. Computer Science Dept., University of London, UK. PhD Thesis.
Schank RC (1982) Dynamic Memory, a Theory of Understanding and Learning in Computers and People. Cambridge UK: University Press.
Shim JP, Warkentin M, Courtney JF, Power DJ, Sharda R, Carlsson C (2002) Past, present, and future of decision support technology. Decision Support Systems, 33 (2): 111-126.
Simon HA (1977) The New Science of Management Decisions. New Jersey, NJ: Prentice Hall.
Sprague RH (1993) A framework for the development of decision support systems. In (Eds. Sprague, R.H. and Watson, H.J.), Decision Support Systems Putting Theory into Practice, (pp. 3-26). New Jersey, NJ: Prentice-Hall International.
Sun R (2001) Artificial intelligence: connectionist and symbolic approaches. In (Eds. Smelser, N.J. and Baltes, P.B.), International Encyclopedia of the Social and Behavioral Sciences, (pp. 783-789). Oxford England: Pergamon/Elsevier.
Tecuci G, Kodratoff Y (1995) Machine Learning and Knowledge Acquisition: Integrated Approaches. London UK: Academic Press.
Teng JTC, Mirani R, Sinha A (1988). A unified architecture for intelligent DSS. In proceedings of the 21st Annual Hawaii International Conference on System Sciences (HICSS-21), (pp. 286-294). Hawaii USA: IEEE Computer Society Press.
Turban E, Aronson J (1998) Decision Support Systems and Intelligence Systems. 5th Edn., New Jersey: Prentice-Hall.
Turban E, Aronson J, Liang T (2005). Decision Support Systems and Intelligence Systems. 7th Edn., New Jersey: Pearson Prentice-Hall.

Viademonte S, Leao BF, Hoppen N (1995) Hybrid model for classification expert system. XXI Latin America Conference on Computer Science, (pp. 639-648). Canela, Brazil.
Viademonte S, Burstein F, Dahni R, Williams S (2001) Discovering knowledge from meteorological databases: a meteorological aviation forecast study. Third International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2001), (pp. 61-70). Munich, Germany. Berlin, Germany: Springer-Verlag.
Viademonte S, Burstein F (2001) An intelligent decision support model for aviation weather forecasting. Advances in intelligent data analysis: 4th international conference (IDA 2001), (pp. 278-288). Cascais, Portugal. Berlin, Germany: Springer-Verlag.
Viademonte S (2004) A Hybrid Model for Intelligent Decision Support: Combining Data Mining And Artificial Neural Networks. Faculty of Information Technology, Monash University, Australia. PhD. Thesis.
Wang J (1994) Artificial neural networks versus natural neural networks: A connectionist paradigm for preference assessment. Decision Support Systems, 11 (5): 415-429.
Weiss SM, Indurkhya N (1998) Predictive Data Mining: A Practical Guide. (Ed, Morgan, M.B.), San Francisco, CA USA: Morgan Kaufmann Publishers, Inc.
Wu X (1995) Knowledge Acquisition from Databases. Norwood, NJ: Ablex Publishing.

5

Taking Decisions into the Wild: An AI Perspective in the Design of i-DMSS

Hamid R. Ekbia

Computer Science Department and The Redlands Institute
University of Redlands, 1200 E. Colton Ave. Redlands, CA. 92373, USA
e-mail: hamid_ekbia@redlands.edu

Artificial intelligence has influenced DSS at different stages of its development, but this influence has followed an intermittent pattern; the most recent manifestation of which is a conceptual gap between the two fields. This chapter seeks to narrow the gap by tracing the development of the two fields from their common origin in the ideas of Herbert Simon to the present time. It demonstrates that while AI has departed from those ideas, DSS has remained largely influenced by them. Following a top-down approach, the chapter examines some of the basic premises of current DSS in order to develop a new conceptual framework. We draw upon recent trends in AI towards situated models to propose an embedded, action-oriented, and improvizational approach to the design of intelligent decision-making support system (i-DMSS), and outline a methodology that would support this framework

5.1 Introduction

Decision support systems (DSS) have evolved significantly during the last four decades. However, their capabilities are still very limited. Elgarah *et al.* (2002), for instance, writing on a project to develop a DSS for urban infrastructure decision-making for the city of Houston, report that they "know of no DSS design methodology suitable for use in such a complex, conflict-filled situation as this." This is an alarming observation in the face of decades of research and practice on DSS. This chapter seeks to suggest remedies to this situation.

The major reason for the current shortcomings of DSS, I argue, has to do with the technocentric nature of the development of these systems. An overview of the history of DSS reveals that its development has been largely driven by changes and innovations in computer technologies such as data and knowledge bases, expert systems, software agents, and more recently Web-based tools. This reveals the dominance of a techno-centric view in DSS development that is also manifested in its relationship to artificial intelligence (AI). Both of these areas were originally influenced by the ideas of Herbert Simon - DSS through Simon's (1960) seminal work in management science and AI through his work, with Allen Newell (1961, 1976), on human problem-solving and on the *Physical Symbol Systems' Hypothesis* (PSSH) -. However, the two areas have followed rather different development paths for a good part of their history, as Simon himself observed many years ago (1987).

As it turns out, the AI community, having discovered the limits and flaws of PSSH, has made serious departures from its underlying premises and assumptions, while the DSS community has remained largely committed to the traditional concept of decision making formulated by Simon - or, at least, it has not seriously questioned some of the basic premises and assumptions of his views -. Given that the DSS community usefully understands AI as "a reference discipline for DSS research" (Goul *et al.* 1992), it might be worthwhile revisiting the relationship between the two fields once again. This is the main thrust of this chapter. Unlike previous similar attempts, however, we will go about this in a top-down fashion. That is, rather than starting with AI tools and technologies that might prove useful for DSS, we start with a study of conceptual developments in AI, examine their implications for DSS, develop a new conceptual framework, and then arrive at a design methodology that would support the framework.

To this end, the chapter traces the development of the two fields from their common origin to the present time. In particular, we draw upon recent developments in AI to propose an embedded, action-oriented, and improvisational approach to the design of intelligent decision-making support system (i-DMSS). The methodology that emerges, best characterized as decision making in the wild, is in line with recent developments in DSS and in software development - *e. g.* active DSS (Shim *et al.* 2002), the WinWin negotiation model (Boehm *et al.* 1995) - and with some of the major empirical findings in the history of the field itself, *e. g.* the superiority of prototyping and evolution-based methodologies (Alter 1978, Alavi 1984, Mahmood and Medewitz 1985).

The chapter continues in the next section with a brief historical outline of the development of AI and DSS, focusing mainly on the ideas of Simon. This is followed by a discussion of recent developments in AI, particularly in the area of situated cognition. Next, we examine the implications of these developments for the conceptualization and design of i-DMSS. This results in a conceptual framework that we then apply to outline a new methodology.

5.2 The Common Origins of AI and DSS

The emergence of AI as a reference discipline for DSS has been previously explored (Goul *et al.* 1992, Eom 1998). Rather than "emergence" however, the relationship between the two areas might be better described as coorigination. For not only did the two disciplines emerge at about the same time (late 1950s), they both have a common beginning in the works of Herbert Simon. Simon is deservedly known as a pioneer of AI and as the founder of decision science as we know it today. His book "The New Science of Management Decision", which brings these two major strands of his thinking and research together, provides a good context for the joint understanding of Simon's views and of the history of these two areas.

In the opening chapter of his book, which deals with the impact of computer technology on the processes of management, Simon discerns three dimensions of disagreement among experts on the degree of this impact: a technological dimension, a philosophical dimension, and a socioeconomic dimension. Roughly speaking, according to Simon, expert opinions vary depending on how small or big they envision the impact. Aligning the first two dimensions, Simon recognizes four possible schools of thought in this respect (see Table 5.1), and characterizes his own position as "fairly extreme along all dimensions" - namely as a technological radical, an economic conservative, and a philosophic pragmatist (1977: p. 6). In line with this position, he then makes the following assertions and predictions about the role of computers in organizations:

I believe that in our times computers will be able to perform any cognitive task that a person can perform. I believe that computers already can read, think, learn, create. I believe that computers and automation will contribute to a continuing, but not greatly accelerated, rise in productivity, that full employment will be maintained in the face of that rise, and that mankind will not find life of production and consumption in a more automated world greatly different from what it has been in the past. (pp. 6–7).

Almost forty years after these prophecies, it is indeed sobering to see how many of them have not yet materialized (*e. g.* full employment and disappearance of poverty and deprivation that he predicts later in his book), and how many others are still debated (*e. g.* rise in productivity, employee empowerment, *et al.*; *cf.* Kling 1996). Of particular relevance to the present discussion, however, is Simon's radical view about the capabilities of computers, especially as they relate to organizational and management processes. To understand the source of this radicalism, we need to study Simon's views of human cognition and of organizational decision-making.

Table 5.1. Simon's classification of views on the impact of computers on management

Philosophical / Technological		Socioeconomical	
		Conservative	Radical
	conservative	Computers are limited in power, and business is done as usual	Computers are limited in power, but there will be plenty of goods and services
	radical	Computers equal humans in terms of capabilities, but business is done as usual	Computers equal humans in terms of capabilities and will replace humans

5.2.1 Cognition as Means-Ends Analysis

As mentioned earlier, Simon's view of human cognition is best represented in his work with Newell on human problem solving. The main idea behind this hypothesis is that, "human thinking is governed by programs that organize myriads of simple information processes - or symbol manipulating processes if you like - into orderly, complex sequences that are responsive to and adaptive to the task environment and the clues that are extracted from that environment as the sequences unfold" (Simon 1977, p. 68). This mentalistic view of human cognition is based on a number of key assumptions in what is now called classical AI. As Agre (1997) points out, mentalism's simple answer to all questions of psychology is: put it in the head. "If agents need to think about the world, put analogs of the world in the head. If agents need to act in situations, put data structures called 'situations' in the heads. If agents need to figure out what might happen, put simulations of the world in the head" (p. 51). In short, the basic method of mentalism is to reproduce the entire world inside the head.

Newell and Simon's idea is an exemplar of the mentalistic view that emphasized problem solving in the abstract. In their work on the computer program GPS (*General Problem Solver*), Newell and Simon (1972) referred to means-ends analysis - that is, the analysis of the difference between what we need and what we have - as the key component of human thought process. In their studies of human problem solving under laboratory conditions, they used "thinking-aloud protocols" in order to tap into their subjects' internal thought processes - a methodology that turned into a pillar of knowledge engineering for years to come - . Simon argued that, "Problem solving proceeds by erecting goals, detecting differences between present situation and goal, finding in memory or by search some tools or processes that are relevant to reducing differences of these particular kinds, and applying these tools or processes" (1977, p. 70). He also emphasized that these tools (or heuristics) are "subject-matter free," in the sense that they apply to any problem that can be cast into an appropriate general form.

In short, Simon viewed cognition as a heuristically guided search activity within an abstract mental space. As we shall see next, it was a short step from this view to the idea that a good part of management decision making can in principle be performed by computers and to a putative methodology that would implement it.

5.2.2 The Automation of Management

Within the near future — perhaps in the next generation — we shall have the technical capability of substituting machines for any and all human functions in organizations. (Simon 1977: 16)

On the basis of the above premises, Simon envisioned the automation of management in the spirit of what had previously happened in factory automation. He assessed this development as a "technological revolution of the decision-making process" (1977, p. 31). To motivate this, Simon classified organizational decisions into two major categories: "programmed" (structured) and "unprogrammed" (unstructured). The first category, which refers to the routine and cut-and-dried decisions of heads and middle managers, belongs in the realm of operations research and its formal and mathematical techniques such as linear programming. The second category, which consists of the basic, once-for-all, and unusually consequential decisions of presidents and top managers, falls into the domain of artificial intelligence (or as Simon sometimes preferred to call "heuristic programming"). In the same manner that humans reach "unprogrammed decisions by reducing them to a series of programmed decisions," according to Simon, executives can similarly do this by following a number of well-defined, but interwoven, phases.

This conceptualization of organizational decision-making gave rise to Simon's well-known four-phase model of decision-making — intelligence, design, choice, and review - which dominated most design methodologies in DSS for years to come (Gorry and Scott-Morton 1971) -. Simon describes the main focus of each of these phases as follows:

Intelligence:	Survey the economic, technical, political, and social environment to identify new conditions;
Design:	Invent, design, and develop possible courses of action for handling situations;
Choice:	Choose among alternative actions already developed to meet an identified problem and already analyzed in terms of their consequences;
Review:	Assess the outcomes of past actions as part of a repeating cycle that leads again to new decisions;

I shall demonstrate later how in practice decision-making processes differ from this conceptualization. But even a superficial examination of current organizational processes would illustrate how far indeed we are from "substituting machines for any and all human functions in organizations." This gap between Simon's predictions and current reality might be indicative of serious flaws in his original conceptualization of thought processes in humans and of decision-making processes in organizations. The task of discovering those flaws has fallen onto the shoulders of the AI and DSS communities, respectively, but the two communities have responded differently to this task.

5.2.3 The Divergent Paths of AI and DSS

Despite all the changes and accomplishments in DSS tools and techniques, the conceptual framework laid out by Simon seems to have largely remained intact, dominating design methodologies in DSS throughout decades. At least, to the best of my knowledge, no one in the DSS community has explicitly and systematically questioned the premises and assumptions behind that framework. AI, in the meantime, has gone through serious challenges and changes that greatly depart from the classical notion of cognition as heuristic problem-solving. This has created a conceptual chasm between the two disciplines that researchers have tried to bridge every once in a while (Goul *et al.* 1992, Eom 1998). Nevertheless, contacts between AI and DSS has largely remained at the level of tools and techniques, partly because of the technocentric character of development in DSS (in contrast to scientific and philosophical aspirations of AI; *cf.* Ekbia forthcoming). This, of course, does not imply that AI has followed a steady and smooth development path. In fact, as different authors have argued and as the following discussion demonstrates, the development of AI has been tumultuous, strained, and stifled with false starts (Bloomfield 1987, Crevier 1993, Agre 1997).

5.3 The Development of AI: From Classical to Situated View

The development history of AI can be roughly divided into four periods dominated by four major approaches, which I am going to call: i) the classical approach, ii) the knowledge-based approach, iii) the connectionist approach, and iii) the situated approach. Although there are overlaps among these, each one is differentiated by basic features and premises (see Table 5.2).

Table 5.2. Major approaches to AI and their key idea

Approach	Main idea
Classical	Symbol manipulation
Knowledge-based	Knowledge
Case-based	Reminiscence
Connectionist	Distributed computation
Situated	Embodiment and embeddedness

5.3.1 The Classical Approach

Classical AI viewed cognition as abstract (physical embodiment is irrelevant), individual (the solitary mind is the essential locus of intelligence), rational (reasoning is paradigmatic of intelligence), and detached (thinking is separated from perception and action) (Smith 1999). Relying on these principles, early AI, as exemplified by PSSH, did in fact make long strides. Systems built upon these were relatively successful in abstract problem solving, but they abruptly failed in dealing with more concrete tasks and domains that apparently seemed mundane and trivial for human beings - *e. g.* recognition of letters of the alphabet, translation between languages, navigation in nonidealized terrains, and so on- .

The failure of the classical approach to tackle these issues and to deliver its promises resulted in a decline of interest in AI research on the part of funding agencies in the early 1980s, leading practitioners to look for practical problems to solve.

5.3.2 The Knowledge-based Approach

As knowledge was conceived to be the key to such practice-oriented endeavor, a new class of artifacts ("expert systems") and a new group of practitioners ("knowledge engineers") appeared on the scene. The task of a knowledge engineer was considered to be threefold (Hayes 1990, p. 201):

- To elicit from an expert - *i. e.*, "a human being whose head contains knowledge"- the knowledge (they intuit) they deploy in dealing with their area of expertise;
- To formalize this knowledge, using a suitable representational language, in a knowledge base;
- To compare the performance of this formalization with that of the expert for the purpose of "correcting" machine behavior - *i. e.*, bringing it closer to the expert's introspective account -.

The practice of knowledge engineers and early success of expert systems bestowed upon AI the respect it was longing for, not only in academia, but also in business, where, according to some accounts, billions of dollars were invested in expert systems for manufacturing, financial services, machinery diagnosis, and so on. But this success was very limited because of the fragility of these systems. Despite apparent sophistication and expertise in specialized areas like medicine, expert systems demonstrated a clear lack of understanding of very basic facts that a human being takes for granted.

Attempts to rectify this situation have largely failed to date, as best exemplified by the Cyc project (Lenat and Guha 1990). Cyc was motivated by the idea that a vast amount of knowledge is the key to intelligence. What we need to achieve human-level intelligence, the creators of Cyc believed, is to provide a machine with enough commonsense knowledge for it to be able to continue the process of knowledge acquisition on its own. Therefore, they embarked on the creation of a huge knowledge base that after a certain point was meant to learn on its own and go "beyond the frontiers of human knowledge." This turned out to be an elusive objective due to, among other things, the vast, implicit, and contextual character of a great deal of human knowledge (Smith 1991). Organizations face similar issues in dealing with knowledge (Blackler 1995).

The knowledge-based approach was concomitant with two other views. One is case-based reasoning, which emphasizes reminiscence rather than knowledge (Schank 1982). The other is the so-called planning view, which considers human activity as fundamentally planned and well thought out. Like the knowledge-based approach, both of these views have faced insurmountable obstacles, however (Suchman 1987, Hofstadter 1995).

5.3.3 The Connectionist Approach

The connectionist approach (re)emerged in the 1980's and captured the imagination of many researchers, including some in DSS. The main feature of this approach was its opposition to explicit forms of knowledge and its emphasis on brain-like architectures, but it remained committed to most of the principles of classical AI. In particular, it retained the notion of mental representation that was central to classical theories of mind. Therefore, despite the initial fervor, connectionism could not distract AI from some of its fundamental premises (Clark 1997, Clancey 1997). This was left to later developments, especially those associated with the situated view of cognition.

5.3.4 The Situated Approach

The situated view reverses many of the assumptions of the previous approaches, especially those of classical AI. That is, it considers intelligence to be embodied (physical embodiment is important), embedded (the immediate natural and social environment matters), action-oriented and largely improvisational (Agre 1997, Clancey 1997, Clark 1997, Smith 1999).

These conceptual shifts in AI approaches (which we shall discuss in more detail shortly) are consequential not only for our understanding of human intelligence but also, I argue, for the design of artifacts such as computerized decision support systems. To be able to articulate these consequences, we also need to follow the development of DSS since its coorigination with AI.

Table 5.3. The development of DSS in relationship to computer technology

Stage	Approximate period	Dominant concept of DSS	Technologies
I	1960s–1970s	Data modeling and problem solving	Databases, MIS
II	1980s	Collaborative and Group Decision Support (GSS)	Knowledge bases, expert systems, EIS
III	1990s	Organizational learning and Knowledge Management	OLAP, data warehouse, data mining
IV	2000s	Web-based and active DSS	Internet, client-server tools, software agents

5.4 The Development of DSS: From Data Modeling to Active Systems

As mentioned at the outset, DSS have significantly evolved during the last few decades. In fact, the DSS community (being commendably self-reflexive) has both summed up and anticipated the development of the field at various junctures. Accordingly, the definition of DSS has changed from support technologies in semistructurred domains (Keen and Scott-Morton 1978) through interactive data models (Sprague and Carlson 1982) and group decision support systems (DeSanctis and Gallupe 1987) to adaptable and domain-specific representational models (Turban 1992) and social DSS (Turoff *et al.* 2002). More recently, Shim *et al.* (2002) have discerned a trend toward the personalization of DSS user interface, the use of Web-based technologies, and ubiquitous computing. These authors have also prescribed the development of active and intelligent systems as a promising path for the future.

A careful examination of DSS development reveals a close parallel between the conceptualizations of DSS and the development of computer technologies and tools (Shim *et al.* 2002). In the era of data-processing and management-information systems (MIS), for example, the emphasis in DSS was on databases and data models. Later, with the advent of expert systems and executive information systems (EIS), the scope of DSS extended to group and corporate levels. Then, the growing interest in knowledge brought about the notion of organizational learning and knowledge management. Most recently, the expansion of the World Wide Web and wireless technologies is giving rise to web-based DSS and to new conceptualizations of decision making from multiple perspectives. As Table 5.3 illustrates, this development has been dominantly bottom-up and technology driven, with the available technical tools giving rise to new conceptualizations.

A comparison of Tables 5.2 and 5.3 illustrates that AI and DSS have followed rather independent development paths, but they have converged at certain points in terms of concepts, methods, and techniques (see Figure 5.1).

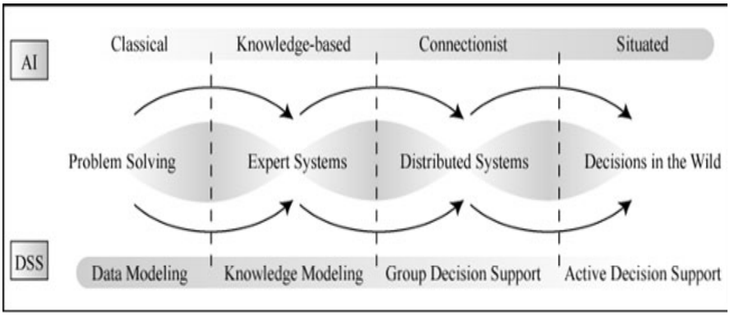


Figure 5.1. The intermittently convergent paths of AI and DSS development

The last such point was coincident with the trend toward knowledge-based DSS. Goul *et al.* captured this junction by anticipating and articulating the above trend in their proposition, which emphasized supporting human decision-making "*by selectively incorporating machine-based expertise in order to deliver the potential of DSS in the knowledge era*" (1992, p. 1268). As we saw, AI has moved beyond the so-called knowledge era to situated interaction, and in order for the two disciplines to maintain their relationship we need to align them once again. The remainder of this chapter is an attempt in this direction.

5.5 A Situated Approach to Design

To go about our goal of alignment, we can follow one of two strategies (see Figure 5.2):

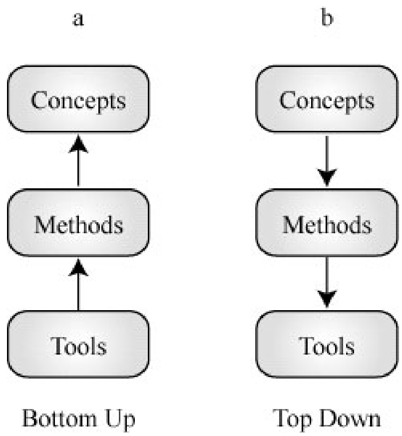


Figure 5.2. Two alignment strategies

- *Bottom-up*: look for the most recent tools and techniques in AI that might be useful for DSS, and then find the methods and concepts that would emerge from them;
- *Top-down*: bridge the conceptual gap between AI and DSS by providing a new conceptualization of DSS, and then find suitable methods and techniques that would support the new conceptualization.

The bottom-up approach has been the strategy of choice in most previous attempts, with certain advantages and disadvantages accrued to it. We follow the top-down strategy here, hoping to demonstrate its advantages throughout the following discussion. In line with the main theme of the current volume, henceforth our discussion will focus on intelligent decision-making support systems (i-DMSS).

For decision support systems to be intelligent, they should be incorporated in their human environment in as seamless a manner as possible. For this purpose, they should be built on the same principles on which human cognition operates. In other words, the design of these systems should follow, to the extent possible, the model of human cognition. Situated AI, we may recall, is based on four key principles that cognition is: i) embodied, ii) embedded, iii) action oriented, and iv) improvizational. The first principle, although very important in the design of AI systems such as robots, is not directly relevant to i-DMSS design, so we do not pursue it further. Let us therefore discuss the other three principles in terms of their significance for the design of i-DMSS.

5.5.1 Embedded Design

The idea of embedded cognition is that meaning can arise only through an intelligent system's direct interaction with the world, where interaction is broadly understood as perception and action. The classical view in AI marginalized perception and action as secondary issues, on the one hand, and considered them as separate processes mediated by a "brain" or central processor, on the other. The situated view, to the contrary, takes perception and action as the centerpiece, and pushes central processing and internal representations of the world to the margin. Furthermore, it considers perception and action as tightly intertwined.

This shift in our understanding of perception and action has important implications for the design of i-DMSS. A key tenet of the problem-solving paradigm in DSS was its emphasis on mental representations of external situations (Simon's "intelligence" phase). According to this view, people deal with the outside situations by building (more or less) faithful models of them in their heads. Therefore, the argument went, all thinking (*e.g.* decision making) consists mainly of the manipulation of these internal models and symbols. Problems are in our heads, as are solutions to problems. According to the situated view, however, problems are not so much in the head as they are in external situations. In other words, what we often have to deal with are "problematic situations," not problems as mental models of those situations. A problematic situation is one that is "disturbed, ambiguous, confused, full of conflicting tendencies, obscure, *et al.*" (Dewey 1938). This means that problems do not present themselves as given, rather "they must be constructed from the materials of problematic situations which are puzzling, troubling, and uncertain" (Weick 1995, p. 9).

An embedded design methodology (not to be confused with "embedded computing") would therefore start with problematic situations rather than with mental problems. It would seek to create a balance between the local, situated knowledge of participants and stakeholders, on the one hand, and global facts and procedures, on the other (Walsham 2001, pp.108–130). The failure to maintain this balance might generate unpredictable side effects in an organization or in the whole society. Walsham (2001) reports a case study of a DSS implemented for the corporate lending process of a large UK bank, where the system was intended to provide lenders with an analysis of a borrower's capacity. The system calculated the probability of the loan defaulting on the basis of similar previous cases in its knowledge base. The parameters of these calculations were weighted to reflect data gathered by knowledge engineers through elicitation of "best practices" from selected "good" managers. In discussing the impacts of the DSS use, Walsham reports that the users "were expected to become dispassionate loan workers, …where loan judgments were seen as adhering to global standards rather than local contingencies" (p. 128). In light of this, Walsham finds it difficult to make a definite assessment of the DSS either in terms of "efficiency, effectiveness and profitability" of the bank or in terms of their profound societal effects - for example, the change in the profile of companies (such as small businesses) that are affected by the system - . Such intricacies in terms of the interaction of local and global knowledge are involved in almost any substantial DSS, and should therefore be taken seriously in an embedded design approach.

5.5.2 Action-oriented Design

Closely related to the above discussion is the issue of human interaction with the world. The classical view of AI considered this as a linear process where the outside world shapes our thoughts, which we then turn into plans that in turn influence the outside world. This is the essence of Simon's four-part scheme of decision-making. However, thoughts and actions are much more intertwined than this linear picture portrays. People do not face a situation as a given, rather they enact and produce the situations of which they are a part. As Garfinkel describes them, "*in the course of a career of actions*, [people] discover the nature of the situations in which they are acting… [T]he actor's own actions are first order determinants of the sense that situations have, in which, literally speaking, actors *find* themselves" (1967, p. 115). For example, Garfinkel found out that jurors do not first evaluate the harm, then allocate blame, and finally choose a remedy. Rather, they first decide a remedy and then settle on the "facts" from among alternatives that justified the remedy. In short, they *retrospectively* justify a decision on grounds other than (or beyond) facts. In other words, if Garfinkel is right, the traditional view of decision-making as one of weighing a set of given alternatives (Simon's "choice" phase) might actually be putting the cart before the horse.

These observations are particularly important in today's uncertain, rapidly changing, and information-laden environments of decision making. The sheer magnitude as well as the inherent uncertainty of data available today turns their full assimilation by decision makers into an unrealistic expectation. Under these circumstances, the whole notion of *planning* (Simon's "design" phase) as a centralized and guided process might be an ungraspable chimera. What people need in most situations is less, not more, information; and what i-DMSS needs to provide first and foremost is the facilitation of action not the accumulation of knowledge. This is in line with Shim *et al.* (2002)'s prescription, originally conceived by Keen and Scott-Morton (1987, p. 121), for active DSS and their emphasis on "screening, sifting, and filtering" of data.

5.5.3 Improvisational Design

This brings us to the third and final aspect of the situated approach to design. In contrasting the planning and situated views of AI, Agre (1997, p. 7) argues that, human activity is "fundamentally improvised; …People conduct their activity by continually redeciding what to do." Given that life is almost routine too, it seems that *human activity is improvized and routine at once*, with improvisations relying on routines. In other words, rather than repeatedly performing the same actions according to preset plans, we utilize our relatively stable relationships with the environment - *e. g.*, in the way we organize furniture, our paper documents, or our computer desktops - as a backdrop for our moment-to-moment interactions with the world.

Brought into the realm of design, this notion of improvization shifts our attention away from products to the *processes* of technical work (Agre 1997, p. 15). Central to this view are "*a willingness to forego planning in favor of acting, an openness to reassembly of and departures from routines, and a preference for process rather* *than structure*" (Walsham 2001, p. 51, *cf.* Weick 1998). Unlike the rigid norms of classical life-cycle models of software engineering, which were often dictated by bureaucratic imperatives, this approach prescribes a more reflective attitude toward software design.

5.6 In Search of a Design Methodology

Having laid out the key conceptual principles of situated design, we now need to explore appropriate methods that would support these concepts. It has long been known that design methods have a direct impact on the success of DSS (Mahmood and Medewitz 1985). Based on the above principles, I propose a five-step method consisting of problem setting, bricolage, coordination, narration, and simulation, a brief discussion of which follows.

5.6.1 Problem Setting

"When we set a problem, we select what we will treat as the "things" of the situation, we set the boundaries of our attention to it, and we impose upon it a coherence which allows us to say what is wrong and in what directions the situation needs to be changed. Problem setting is a process in which, interactively, we name the things to which we will attend and frame the context in which we will attend to them" (Schön 1983: 40).

The first step is problem setting, the purpose of which is to turn a problematic situation into a problem. This is an ongoing, iterative, and reversible process during which the actors try to achieve a common understanding of the situation as a set of relevant interrelated "things." As Schön describes above, this mainly consists of naming the objects (or even the situation as a whole) and setting the boundary of attention. The product of this step would be a "laundry list" of main objects and a schematic outline of the major issues, constraints, and concerns.

5.6.2 Bricolage

Once the objects are named, the next step would be to assemble them in an improvizatory manner. We should keep in mind that the relevant "things" are not merely the objects of expert knowledge or combinations of already existing pieces of technology - hardware, software and facilities - but also of "of appropriate work practices, skills, training, communications …" (Büscher and Morgensen 1997, p. 79). Since part of this cannot be formalized (in the sense of encoding and storing in a knowledge base), various representation schemes should be used to capture and preserve the components - *e. g.*, database tables, knowledge base (KB) rules, frames (schemas), hypertext, images, photos, *et al.* -. The participatory and inclusive character of the assembly phase cannot be overemphasized, as each particular situation involves its own local contingencies.

5.6.3 Coordination

Following the initial assembly of the bricolage phase, a process of filtration and refinement is needed to eliminate or reduce redundancies, mismatches, and conflicts. An explicit attempt should also be made at this stage to transform, to the extent possible, informal representations to formal ones. Ideally, the goal of this stage is to develop an agreed-upon *model*, although in practice this might be difficult to achieve. There is always a residue of local, implicit, and informal knowledge that cannot be formalized and needs to be presented in an informal manner. The next phase (narration) addresses this issue.

5.6.4 Narration

The purpose of narration or story-telling is to give structure and meaning to an otherwise incoherent ensemble of data, objects, models, *et al.* One of the problems of using DSS effectively is the complexity of models and the challenge that this poses to users in terms of comprehension and accessibility. Narration can help users make sense of the models using tools that could be formal or informal, mathematical or descriptive, textual or visual, and so on.

5.6.5 Simulation

Simulations have recently attracted considerable attention among social and organizational scientists. Simulations are useful not only because they provide the opportunity for controlled experimentation - *i. e.* playing around with, and observing the impact of, parameters without affecting the real world -, they are also useful because they "*enable the observation and recording of the background of planning, decision-making, and evaluation processes that are usually hidden*" (Dörner 1996, pp. 9–10). In short, simulations can be conceived as "social labs" that provide an effective way of learning different from the known alternative of *learning by doing* (Bousquet *et al.* 1999).

Although simulations can take the form of mathematical modeling, what is usually intended by the term is either simulation *in ludo* (by people) or *in silico* (by computers), or a combination thereof. The first type of simulation normally takes the form of *role-playing games* (RPG), where people are assigned roles (often different from their real-life status) and given the opportunity to make decisions and to observe the short-term and long-term consequences of those decisions. This helps participants arrive at a shared representation of the problem, which then facilitates the process of collective decision-making. Bousquet *et al. (*1999), who report on the application of RPGs in different projects, describe their various uses in training, research, and policy making. Becu *et al.* (2003) have noticed the usefulness of simulations as "very efficient communication media."

The crucial point is the experimental and, preferably, visual nature of this phase. Whatever the form, the purpose is to give users a chance to experiment with ideas, to understand the consequences, to be able to associate with others, and so on.

5.6.6 Retrospection as Validation

This is probably the most challenging phase, and its purpose is for actors to relive and review the decision-making process by trying to explain it to others. In this manner, retrospection is utilized as a means of validating, legitimating, and making the process transparent. In this fashion, issues of uncertainty will be at least partly handled. Needless to say, this whole process is iterative and incremental, as people might discover in the act of retrospection that critical aspects are missing, conflicts are still outstanding, goals are not achieved, and so on.

In sum, the methodology outlined above supports the conceptual principles laid out earlier. The emphasis in problem setting and bricolage give a strongly embedded flavor to this methodology, narration and retrospection highlight its improvisatory aspect, and coordination and simulation introduce a crucial active component to the process. Technically, this method is also strongly supported by emerging technologies such as those suggested by Shim *et al.* (2002) - *e. g.*, web and networking technologies, ubiquitous computing, and enterprise information systems (EIS) -. This methodology is also in line with earlier empirical findings in the history of DSS - *e. g.* the advantages of prototyping (Alavi 1984) and of evolutionary methods over other alternatives (Alter 1978, Mahmood and Medewitz 1985) -, and finally, with recent developments in software engineering and i-DMSS - *e. g.* Boehm's Theory W (Boehm *et al.* 1995).

5.7 Conclusion

What is missing in the design of many contemporary IS is an effective link between the planned, automated decision process and all those tacit aspects, such as the because-of motives, or past experience, which give meaning to the development and implementation of a decision. This is why automated procedures tend so often to be underutilized, for they do not match changing circumstances, badly mimic the know-how of even a novice, feel unnatural and clumsy, seem to lack meaning and be out of context, and are full of loopholes which have to be filled by [improvised] human intervention. (Ciborra 1996, p. 375; in Walsham 2001)

AI and DSS have both undergone dramatic changes since their common inception about half a century ago. In the course of this development, the two fields have mutually informed each other, but most of their dealings have taken place at the technical level. The DSS community, in particular, has always stayed "on top of" the latest technical developments in AI and other computer technologies. However, as Ciborra contends in the above quote, this has not always resulted in the most effective, meaningful, useful, and user-friendly systems. This chapter is an attempt to remedy this situation by aligning DSS with recent developments in AI. Following a top-down approach, we began at the conceptual level and derived basic design principles that, in turn, led us to methodological (and partly technical) levels. The outcome of this study is a perspective on the design of iDMSS with the following highlights:

- By introducing activities into the picture, this perspective emphasizes the process of decision-making rather than its product;
- By downplaying mental models, it reduces the cognitive load of deliberation on decision makers;
- By starting from the external situation, it makes it more likely and probably easier for multiple decision makers to arrive at a common representation of the problem (which is arguably a major step toward consensus-building);
- By incorporating improvisation, it increases the likelihood of more reflective and adaptable outcomes.

These are the crucial features of the conceptual and methodological approach presented here. In summary, the big lesson of recent developments in AI for the design of i-DMSS is that a good deal of human cognition takes place not in individual, detached, and disembodied brains, but in the contextual interactions among embodied human beings - or, put metaphorically, cognition takes place in the wild (Hutchins 1995) -. This is the lesson that the DSS community should take most seriously. From the boardroom to the floor and from the headquarters to the field - this should be the motto of future i-DMSS designers -.

Acknowledgements

This work was supported in part by funds provided by the U.S. ARL and the U.S. ARO under grant number DAAD19-01-1-0793. The author would like to thank the Redlands Institute for the continuing support, Paul Burgess and Frank Davenport for very useful discussions, and Lindsey Devlin for help with the diagrams.

References

Agre PE (1997) Computation and Human Experience. Cambridge University Press.
Alter S (1978) Development patterns for decision support systems. MIS Quarterly, September, 33-42.
Alavi M (1984) An assessment of the prototyping approach to information systems development. Communications of the ACM, 27(6):556-563.
Becu N, Bousquet F, Barreteau O, Perez P, Walker A (2003) A methodology for eliciting and modelling stakeholders' representations with agent-based modelling. In D. Hales, B. Edmonds, E. Norling & J. Rouchier (Eds.), Multi-agent-based simulation iii - 4th international workshop (Vol. Lecture Notes in Artificial Intelligence 2927, 131-148. Melbourne, Australia: Springer.
Blackler F (1995) Knowledge, knowledge work and organizations: an overview and interpretation, organization studies, 16(6), 1021-1046.
Boehm BW, Bose P, Horowitz E, Lee MJ (1995) Software Requirements Negotiation and Renegotiation Aids: A Theory-W Based Spiral Approach. Proceedings of ICSE-17, 243 - 253.
Bloomfield BP (1987) The Question of Artificial Intelligence. London, UK: Croom Helm.

Bousquet F, Le Page C, Mullon C, Weber J (1999) An environmental modeling approach : the use of multi-agent simulations. In F. B. a. A. Weill (Ed.), Advances in environmental modeling 113-122, Elsevier.

Büscher M, Morgensen PH (1997). Mediating Change: Translation and Mediation in the Context of Bricolage. In: T. McMaster *et al.* (Eds.), Facilitating Technology Transfer Through Partnership: Learning from Practice and Research. London: Chapman Hall, 1997.

Ciborra CU (1996) Improvisation and information technology in organizations. In J.I. DeGross, S. Jarvenpaa and A. Srinivasan, eds. Proceedings of the 17th International Conference on Information Systems. Baltimore: ACM.

Clancey WJ (1997) Situated Cognition: On Human Knowledge and Computer epresentations. Cambridge, UK: University of Cambridge.

Clark A (1997) Being There: Putting Brain, Body, and World Together Again. Cambridge, MA: MIT Press/Bradford Books.

Crevier D (1993) The Tumultuous History of The Search for Artificial Intelligence. New York: Basic Books.

DeSanctis, G, Gallupe BA (1987) A foundation for the study of group decision support systems. Management Science, 33(5): 589-609.

Dewey J (1938) Logic: The Theory of Inquiry. Carbondale, IL: Southern Illinois University Press.

Dörner D (1996) The Logic of Failure: Recognizing and Avoiding Error in Complex Situations. Cambridge, MA: Perseus Books.

Ekbia HR (2005). Artificial Dreams: The Human Quest for Non-Biological Intelligence. (in press).

Elgarah W, Courtney JF, Haynes JD (2002) A dialectical methodology for decision support systems design, 35th Annual Hawaii International Conference on System Sciences. Big Island, Hawaii, USA: IEEE Computer Society, Vol. 256.

Eom SB (1998) An overview of contributions to the decision support systems area from artificial intelligence, Proceedings of the Americas Conference on Information Systems. Baltimore, USA.

Gorry GA, Scott-Morton MS (1971) A framework for management information systems. Sloan Management Review, 13(1): 50-70.

Goul M, Henderson J, Tonge F (1992) The emergence of artificial intelligence as a reference discipline for decision support systems research. Decision Sciences 23: 1263-1276.

Garfinkel H. (1967). Studies in ethnomethodology. Englewood Cliffs, New Jersey: Prentice-Hall.

Hayes PJ (1990) The Naive Physics Manifesto. In M. A. Boden (Ed.), The Philosophy of Artificial Intelligence (pp. 171–205). Oxford, UK: Oxford University Press.

Hofstadter DR (1995) Fluid Concepts and Creative Analogies. New York: Basic Books.

Hutchins E (1995) Cognition in the Wild. Cambridge, MA: The MIT Press.

Keen PGW, Scott-Morton MS (1978) Decision Support Systems: An Organizational Perspective. Addison-Wesley Series on Decision Support. Reading, Massachussets: Addison-Wesley Publication Company.

Kling R (1996) Computerization and Controversy: Value Conflicts and Social Choices. San Diego, CA: Academic Press.

Lenat DB, Guha RV (1990). Building Large Knowledge-Based Systems. Reading, MA: Addison Wesley.

Mahmood MA, Medewitz JN (1985) Impact of design methods on decision support systemws success: an empirical assessment. Information & Management, 9(3): 137-151.

Newell A, Simon HA (1961) GPS, A Program that Simulates Human Thought. Lernende Automaten, H. Billing (ed.) 109-124, Munich: R. Oldenbourg KG.

Newell A, Simon HA (1972) Human Problem Solving. Englewood Cliffs, N.J.: Prentice-Hall.
Newell A, Simon HA (1976) Computer science as empirical inquiry: symbols and search. Communications of ACM, 19(3): 113-126.
Schank RC (1982) Dynamic Memory: A Theory of Learning in Computers and People. Cambridge, UK: Cambridge University Press.
Schön DA (1983) The reflective practitioner: How Professionals Think in Action. New York: Basic Books.
Shim JP, Warkentin M, Courtney JF, Power DJ, Sharda R, Calsson C (2002) Past, present, and future of decision support technology, Decision Support Systems 33(2): 111-26.
Simon HA (1960) The New Science of Management Decision. Englewood: NJ: Prantice-Hall.
Simon HA (1977) The New Science of Management Decisions. New Jersey, NJ: Prentice Hall.
Simon, HA (1987) Two heads are better than one: the collaboration between AI and OR. Interfaces, 17(4): 8-15.
Smith BC (1991). The Owl and the electric encyclopedia. Artificial Intelligence, 47(1–3): 252-288.
Smith BC (1999) Situatedness. In MIT Encyclopedia of Cognitive Science (MITECS). Cambridge, MA: MIT Press.
Sprague RH, Carlson ED (eds.) (1982) Building Effective Decision Support Systems. Englewood Cliffs, NJ: Prentice-Hall.
Suchman L (1987) Plans and Situated Actions. Cambridge, U.K.: Cambridge University Press.
Turban E (1992) Decision Support and Expert Systems. Englewood Cliffs, NJ: Prentice-Hall.
Turoff M, Hiltz SR, Cho HK, Li Z., Wang Y (2002) Social decision support systems. Paper presented at the 35th Annual Hawaii International Conference on System Sciences, Big Island, Hawaii, 2002.
Walsham G (2001) Making a World of Difference: IT in a Global Context. West Sussex: UK: John Wiley & Sons.
Weick KE (1995) Sensemaking in Organizations. Thousand Oaks: Sage Publications.
Weick KE (1998) Improvisation as a Mindset for Organizational Analysis, Organizational Science, 9(5): 543–555.

6

Development Processes of Intelligent Decision-making Support Systems: Review and Perspective

Alexandre Gachet[1] and Pius Haettenschwiler[2]

[1] Department of Information Technology Management, University of Hawaii at Manoa, 2404 Maile Way, A202, Honolulu, HI 96822, USA. gachet@hawaii.edu
[2] Department of Informatics, University of Fribourg, Faucigny 2, 1700 Fribourg, Switzerland. pius.haettenschwiler@unifr.ch

A large amount of research on the development of decision-making support systems (DMSS) focuses on organizational issues, technical issues, or both kinds of issues at the same time. Whereas it is widely recognized that these two categories represent the dominant sources of issues that DSS builders had to overcome in the past, a third category, knowledge-management issues, gradually surfaces. Bolstered by advances in information technology in general, and artificial intelligence in particular, the field of knowledge management increases the number of development issues previously dealt with partly from an organizational perspective and partly from a technical perspective, but rarely as a perspective of its own. This chapter reviews nine DMSS development processes and methodologies according to their focus on organizational issues, technical issues, or both. It then proposes an innovative framework for the design and development of DMSS with particular attention on knowledge management issues and their relation with organizational and technical issues. It finally revisits two existing development processes according to this framework.

"There will always be a tension – ideally a creative one – in the DSS field between Decision and System. The link is Support. The quality of the support we can provide managers depends on our understanding of both decision-making and system building."

-- Peter G. W. Keen, foreword to John L. Bennett, *Building Decision Support Systems* (Addison-Wesley, 1983), p. vi.

6.1 Introduction

Decision-making support systems (DMSS) encompass a collective set of computerized support tools, including decision support systems (DSS), executive information systems (EIS), expert systems (ES), knowledge-based systems (KBS), and other standalone systems (Forgionne *et al.* 2002). In the context of this chapter, however, the expressions DMSS and DSS are used interchangeably.

Finding appropriate DSS development processes and methodologies is a topic that has kept researchers in the decision support community busy for at least the past three decades. For the purpose of our study, we adopt as a definition of the term *process* "the means by which people, procedures, methods, equipment, and tools are integrated to produce a desired end result" (Paulk 1998). Processes are sometimes included in a larger body of knowledge commonly referred to as a methodology. Accordingly, we adopt as a definition of the term *methodology* "[a] body of methods, rules, postulates, procedures, and processes that are used to manage a software engineering project" (Burback 1997).

Inspired by Gibson and Nolan's curve (Gibson and Nolan 1974, Nolan 1979), it is fair to contend that the field of DSS development is reaching the end of its expansion (or contagion) stage, which is characterized by the proliferation of processes and methodologies in all areas of decision support. Studies on DSS development methodologies conducted during the last fifteen years (*e. g.* Arinze 1991, Saxena 1992) have identified more than thirty different approaches to the design and construction of decision support methods and systems (Marakas 1999). Interestingly enough, none of these approaches predominate and the various DSS development processes usually remain very distinct and project specific. This situation can be interpreted as a sign that the field of DSS development should soon enter into its formalization (or control) stage. Therefore, the objective of this chapter is not to come up with new processes and methodologies, but rather to focus on the controlled integration of the existing solutions in a unified body of knowledge that will ultimately lead to the maturity stage.

For the purposes of this chapter, we consider that development processes presented in the DSS literature can be broken down into three broad categories: *(a)* processes focusing on the "decision-making" aspects of the DSS, *(b)* processes focusing on the "system-engineering" aspects of the DSS, and *(c)* processes trying to integrate "decision-making" and "system engineering" aspects in a unified procedure.

Historically, processes focusing on the decision-making aspects of the DSS tend to deal with the organizational perspective of a decision-making process, whereas processes focusing on the system-engineering aspects tackle the technical perspective of the underlying system. A DSS being intrinsically about decision-making *and* about a system, the development processes trying to integrate both aspects in a unified procedure obviously seem the best choice. Nevertheless, advances in the field of knowledge management and knowledge engineering, especially in Artificial Intelligence (AI) (Eom 1998), did not yet find a clear position in this organizational/technical continuum, despite recent efforts to formalize the development processes of knowledge management systems (Tiwana 2002) and knowledge-based DSS (Power 2002)[2]. As a consequence, most of the DSS research remains highly focused on organizational and/or technical perspectives.

The purpose of this chapter is two-fold. In the first part, it offers a review of existing DSS design and development processes. The goal of this review is to give the reader a thorough understanding of the past and ongoing research in DSS development, and to discuss the weaknesses of existing development processes with regard to AI and knowledge management. The second part of the chapter introduces an innovative framework for the design and development of DSS, explicitly taking into consideration the knowledge-engineering aspects of modern decision making. To conclude, the chapter revisits two existing development processes according to the proposed framework and discusses future work.

6.2 Review of DSS Design and Development Processes

DSS design and development processes have been the subject of several comparative studies in the past. Each of these studies categorizes the processes according to varying criteria. For example, Arinze (1991) surveys and analyzes ten DSS methodologies by paradigm, structure, and orientation. For the purpose of his review, the paradigm refers to the models underlying the methodologies (decision driven, process driven, data driven, or systemic). The structure indicates the approach used for guiding the development process (stage or contingency). The orientation involves the developmental guidelines adopted by DSS researchers (normative or descriptive). Arinze's survey leads to a contingency model for DSS methodology selection.

Another example of such comparative studies is provided by Arnott (1998), who analyzes twelve DSS methodologies and DSS development use cases with a strong focus on DSS evolution. He proposes a framework based on the aetiology, lineage, and tempo of evolution. For the purpose of his framework, aetiology refers to the causes of evolution (exogenous or endogenous triggers), "lineage [refers] to whether evolution occurs within an application or between applications, and tempo relates to the pattern of evolution over time" (continuous evolution, punctuated equilibrium, or quantum evolution). Arnott claims that his framework clarifies the nature of DSS evolution and "may help systems analysts predict what may happen next in the development processes and help them in deciding which techniques and tools are likely to succeed with each class of evolution."

For the purpose of our research, we offer a different review and analyze nine DSS development processes with regard to their main focus. Some processes clearly focus on "decision making", others on "system engineering", and others on both. The understanding of the differences between the three categories is then used to propose an integrative, tripartite development framework. Table 6.1 briefly summarizes the reviewed methodologies.

---

[2] As "knowledge-driven DSS are built using explicit, structured knowledge" (Power 2002, p. 143), they represent a subset of knowledge management systems (KMS). KMS researchers also consider unstructured and tacit knowledge in their development processes (*e.g.*, Tiwana 2002).

Table 6.1. The reviewed DSS development methodologies

Methodologies	Focus	Highlights	Reference
Functional mapping	Decision making	Maps the functions of a DSS with the organizational units of the company	Blanning (1979)
Decision graph	Decision making	Used to create the decision model and to identify pertinent decisions at the operational, tactical, and strategic levels of the organization	Martin (1982)
Decision-oriented DSS development process	Decision making	Explicitly distinguishes between the substance (what) and the procedure (how) of the decision-making process	Stabell (1983)
System development life cycle	System engineering	Top-down design philosophy, made up of several phases, each comprised of multiple steps	Sage (1991)
Prototyping	System engineering	Relies on partial versions of a program to aid in designing the final product	Courbon *et al.* (1979, 1980)
End-user development	System engineering	Used by developers who are not trained IS professionals	Alawi and Weiss (1985)
Design cycle	Integrated focus	Evolutionary design insisting on the ties necessary between the decision support functionalities and their actual implementation	Keen and Scott Morton (1978)
DSS development phases	Integrated focus	Driven by user needs	Sprague and Carlson (1982)
Decision support engineering	Integrated focus	Defined as a process of negotiation between the decision maker and the builder	Saxena (1991)

The selected methodologies have been introduced by their respective authors between 1978 and 1991. Research on DSS development did not end in 1991, though. Researchers and practitioners continue to propose new or modified solutions on a regular basis (for example, Holsapple and Whinston 1996, Marakas 1999, Power 2002, Gachet and Sprague 2005). However, most of these new methodologies remain strongly inspired by the works of the previous generation and we decided to present in this review solutions that have been widely acknowledged by the community.

6.2.1 Processes with a Focus on "Decision-making"

DSS development processes focusing on decision-making were particularly popular during the early years of DSS research. We selected three processes introduced by their respective authors between 1979 and 1983, and that are still regularly quoted in the modern DSS literature. Processes are presented in chronological order.

6.2.1.1 Functional Mapping

The *functional mapping* methodology – sometimes referred to as *functional category analysis* (Marakas 1999) – was introduced by Blanning (1979) as a DSS design approach mapping the functions of a DSS with the organizational units of the company. Blanning identifies six functions, summarized in Table 6.2. This methodology clearly identifies the responsibilities and/or benefits of the various organizational units vis-a-vis the DSS.

Table 6.2. The functions of a DSS

	Functions
a.	*Selection* of data from a database
b.	*Aggregation* of data into summary statistics such as totals, averages, frequency distributions, groupings, *et al.*
c.	*Estimation* of the parameters in a probability distribution
d.	*Simulation* to calculate the anticipated outcomes or consequences of proposed decision alternatives
e.	*Equalization* to calculate alternatives whose consequences will meet certain consistency conditions
f.	*Optimization* to determine decisions that will maximize or minimize a single measure of performance or cost without violating constraints on other such measures

Figure 6.1 reproduces an example of functional mapping from Blanning (1979), for a DSS constructed to determine the financial and logistical impact of proposed changes in the product line of a manufacturing corporation.

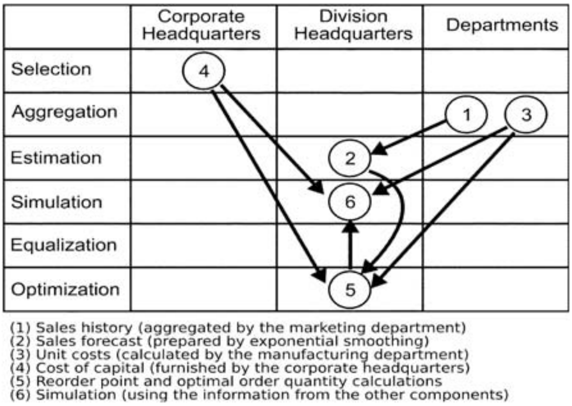


Figure 6.1. An example of functional mapping (from Blanning 1979)

The author claims that the functional mapping methodology can help reduce the development time of DSS projects "by identifying existing modules that could be modified for inclusion in the system" (p. 90). The mapping can also help coordinate the project "by making explicit the responsibilities for modifying the modules" (p. 90), and can help reduce the fragmentation of information systems efforts by "identifying existing sources of information that may contribute to the DSS" (p. 90).

6.2.1.2 Decision Graph

The *decision graph* methodology was introduced by Martin (1982) as a modification of the descriptive system dynamics model. This methodology "emphasizes graphic rather than computer simulation results; changes terminology from an engineering to a decision oriented context; and allows the use of a standard computer template" (p. 17).

Martin's methodology is purely graphical and uses symbols inspired by system dynamics structures (Figure 6.2).

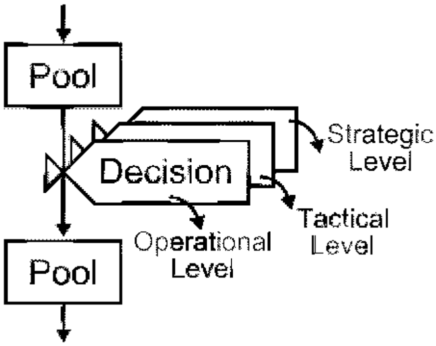


Figure 6.2. The rate of flow of resources between pools is determined by a decision

Decision graphs are used to create the decision model, with the purpose of identifying pertinent decisions. "These decisions are divided into sub-decisions at the operational, tactical and strategic levels of management. Once these sub-decisions are isolated, data elements are identified by observation, interview, deduction or a combination of the three" (p. 21). In this model, pools are accumulations of resources and decisions determine the rate by which resources flow from one pool to another.

6.2.1.3 Decision-oriented DSS Development Process

Stabell (1983) defined the *decision-oriented DSS development process* in reaction to the technocentric, system-oriented development methodologies proposed at the beginning of the 1980s. According to his view, it is "the *decision* in the concept that should define the unique context of DSS, and the *decision* should have implications for the why, how, and what of building such systems" (p. 223). Otherwise, the limited role of the 'D' in DSS leads to ill-understood systems that often have unanticipated impacts (Keen 1980b). According to the author, the distinction between substance (*what* is used during decision making) and procedure (*how* decisions are made) is important, because the development effort is easily led to focus exclusively on the substance of the decision situation.

To attach more attention to the procedure, Stabell's development process relies on three interrelated activities that he collectively labels *decision research*:

- **Data collection**, including data on current decision-making using various techniques (interviews, observation, questionnaires, and historical records);
- **Descriptive modeling**, establishing a coherent description of the current decision process;
- **Normative modeling**, specifying a norm for how decisions should be made.

The other phases of the DSS design and development (such as functional specifications, implementation, monitoring, and evaluation) follow the decision research. Emphasis in this decision-oriented, normative methodology is placed on changing the existing decision process to increase decision-making effectiveness. According to the author, “the development of a DSS should be viewed as an attempt to focus greater attention on the decision-making aspects of the task of managers. The D in DSS is thus used in a prescriptive sense” (p. 230).

6.2.1.4 Discussion

It may seem strange today that some researchers exclusively focused on the “decision-making” part of a DSS, without consideration of system-engineering aspects. After all, it is commonly accepted nowadays that “developing design specifications for a DSS from its requirements specifications involves a cognitive process that maps decision support needs to the tools/technologies available for building it” (Saxena 1991 p. 99).

To understand this apparent inconsistency, it is important to put these processes back into their original context. Builders of early systems - for example, Scott-Morton (1971) - had to deal with complex hardware/software infrastructures and had to overcome significant technological obstacles to create their DSS. This is why researchers like Sprague and Carlson (1982) came up with the concept of a DSS generator, or Bonczek *et al.* (1982) with the concept of a general problem processing system (GPPS). The purpose of such tools was to provide an integrated set of capabilities to build specific DSS quickly and easily[3].

Such generators led part of the DSS community to believe that it was suddenly provided with “a variety of ready-to-use methods and tools (mainly problem-solving and data-management techniques) allowing the efficient ’assembly’ of DSS” (Angehrn and Jelassi 1994, p. 268). If it is true that many of these ready-to-use tools helped to solve original system engineering problems (mostly related to data management, report preparation, inquiry capability, modeling language, graphic display commands, and financial and statistical analysis subroutines), some researchers failed to understand that new advances in computer science and

3 For a complete discussion of DSS generators in the context of DSS development, see (Gachet 2003).

information technology were not only solving existing problems, but also bringing new opportunities (in particular in the field of knowledge management) requiring a new level a mastery in system and software engineering.

Despite these limitations, DSS developers can still draw several lessons from these decision-oriented processes. The focus on decision making naturally leads to strong foundations for the organizational aspects of the DSS and valuable guidelines for the design of the knowledge base of the DSS, that is to say the repository storing information in an organized and structured way (*e. g.*, data, models, rules of inference, cases, or ontologies). In other words, the processes described in the previous sections can help answer two fundamental questions:

- What kind of data and information is needed in the knowledge base?
- What kind of operations should be available to manipulate this data or information?

As will be shown in the second part of this chapter, these two questions have an important impact on the structure of the proposed framework for developing DSS.

6.2.2 Processes with a Focus on "System Engineering"

Development processes focusing exclusively on the "system engineering" part of DSS are rarely specific to decision support systems. They usually borrow their concepts from the system and software engineering fields. We selected three system and software development methodologies often mentioned in the DSS literature.

6.2.2.1 System Development Life Cycle (SDLC)

The *system development life cycle*, or *SDLC*, portrays the process of developing information systems through investigation, analysis, design, implementation and maintenance. The SDLC is characterized by a top-down design philosophy, made up of several phases, each comprised of multiple steps.

The main strength of the SDLC is its ability to make the design process sequential and highly structured. Quite ironically, this strength precisely makes it inadequate for the design and development of DSS. Marakas (1999) points out that "the use of the SDLC presumes that the structure of the problem to be solved or the nature of the problem context in which the DSS will operate is fully known and identifiable prior to the initiation of the design phase" (p. 396), which is simply not true for systems supporting semistructured problems.

Nevertheless, the SDLC remains useful as a reference model and as an idealized abstraction of a more realistic DSS development process. For example, the DSS design and development life cycle is a methodology proposed by Sage (1991) as a phased life-cycle approach to DSS engineering. Its basic structure is very close to the SDLC methodology. However, the author tries to avoid the drawbacks of the SDLC by embedding explicit feedback loops in the sequential life cycle, and by promoting prototyping (see next section) during system implementation, in order to meet the iterative requirements of a DSS development process. The seven steps of this methodology are illustrated in Table 6.3.

Table 6.3. Phases of the DSS design and development life cycle

Phases
1. Identify requirements specifications
2. Preliminary conceptual design
3. Logical design and architectural specifications
4. Detailed design and testing
5. Operational implementation
6. Evaluation and modification
7. Operational deployment

6.2.2.2 Prototyping

Prototyping is an iterative process of systems development in which requirements are converted to a working system that is continually revised through close work between analysts and users. In software engineering, prototyping is described as a process in which partial versions of a program are created to aid in designing the final product. This process facilitates the identification of required functionality during the analysis and design phases.

Historically, one of the first DSS development methodologies based on prototyping is the evolutive approach defined by Courbon *et al.* (1979, 1980) as a "methodology based on the progressive design of a DSS, going through multiple as-short-as-possible cycles, in which successive versions of the system under construction are utilized by the end-user." Table 6.4 summarizes the fours majors steps of this process.

The advantages of prototyping are quite obvious: faster development of systems, lower risk, better communication between developers and users, and better user understanding of the system during the development process. Nevertheless, prototyping also suffers from several drawbacks. Alavi (1984) notes that prototypes are difficult to manage and control, and that it is difficult to prototype large information systems (which DSS often are). According to Marakas (1999), the prototyping process does not give enough attention to detail with regard to the development of comprehensive documentation, and can increase the likelihood that system maintenance will be more difficult than in a comparable SDLC-based system.

Table 6.4 Steps of the evolutive approach

Steps
1. Identify an important subproblem
2. Develop a small but usable prototype to assist the decision maker
3. Refine, expand, and modify the system in cycles
4. Evaluate the system constantly

6.2.2.3 End-user Development

The concept of end-user development – sometimes referred to as shadow IT – refers to people developing software applications for themselves or for others even though they are not trained IS professionals (Kreie *et al.* 2000). End-user development is rapidly gaining popularity in the DSS field. In many large enterprises, end-user development can account for 10 to 80 percent of the size of the total official IT staff

(McNeese *et al.* 2004). Several reasons explain this trend. On the one hand, the advances in information technology made microcomputer hardware relatively cheap, but quite powerful. On the other hand, traditional software packages such as spreadsheets integrate in an intuitive way the definition of reports, graphs and tables in one same application. Another factor explaining the growth of end-user computing is the fact that users are not involved enough in the implementation of the DSS. Therefore, end-user development "[fills] in the blanks left by IT, such as reporting, specialized modeling, and data capture from external sources" (Raden 2005).

The most typical use of end-user development in the field of DSS consists in relying on a spreadsheet package instead of IT-sponsored business intelligence efforts. Even though this autonomous working method is fast and portable, has inexpensive start-up costs, takes advantage of the shallow learning curve of spreadsheets, increases user satisfaction, and avoids communication problems and delays when dealing with the IS department, research shows that end-user computing remains of poor quality (Kreie *et al.* 2000), is poorly managed and easily gets "out of control" (Alavi and Weiss 1985). Raden (2005) identifies other negative implications of end-user development, including wasted time and investment in long-term support and maintenance, inconsistent business logic and approaches between the various applications, inefficiencies, and barrier to enhancement.

6.2.3 Integrated Focus on "Decision-making" and "System Engineering"

The previous two sections reviewed some development processes focusing either on "decision making" or on "system engineering". Whereas processes focusing on decision-making aspects deal with organizational issues and provide guidelines for the design of the knowledge base of the DSS, processes focusing on system engineering should provide guidelines to implement and maintain this knowledge base. This section reviews design processes trying to integrate both focuses in one unifying methodology. Central to most of these processes is the stated necessity to adopt an evolutionary (Moore and Chang 1983), middle-out (Hurst *et al.* 1983), or adaptive design, continually moving back and forth between the decision support needs of the users and the ways to implement them into a coherent system (Keen 1980a). We selected three development methodologies introduced by their respective authors between 1978 and 1991. Methodologies are presented in chronological order.

6.2.3.1 Design Cycle

The design cycle, introduced by Keen and Scott-Morton (1978), can be considered as the ancestor of most of the other DSS development processes. It is widely cited in the DSS literature and is still considered as an authoritative model. The global cycle is shown in Figure 6.3. The boxes identifying decision-making support steps and system engineering steps have been added for the purpose of this review.

Keen and Scott Morton insist on the close ties necessary between the decision support functionalities and their implementation in the actual system. They criticize the "tendency for the design team to focus on the system as an artifact and lose sight of what it is to accomplish" (p. 180). In their view, it is not enough to define a DSS "in terms of inputs and outputs" and to "specify particular data and reports." In other words, it is "essential to emphasize usage of the DSS and what it does rather than what its technical characteristics are" (p. 182).

Their view of evolutionary design is completely in line with the prototyping methodology introduced in the previous section. A system that is usable and useful should be designed and delivered as soon as possible, but it should also be flexible enough to allow rapid extension and addition of routines.

The design cycle also includes aspects typically found in system and software engineering. For example, the separation of the interface from the system functionalities (called "imperatives" in Figure 6.3) represents a traditional design pattern facilitating the addition of routines and the modification or extension of existing ones over time.

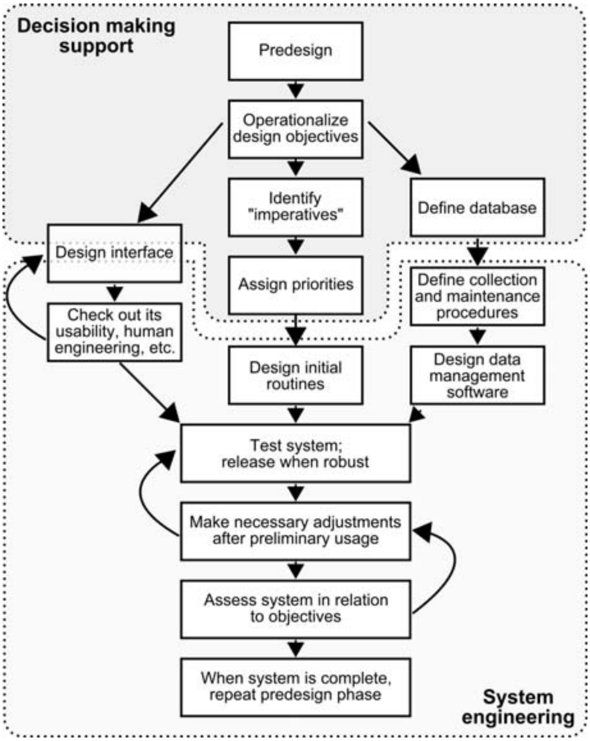


Figure 6.3. The design cycle

One weakness of this methodology is that it makes the system-engineering part of the process linearly dependent on the decision support design part. As we will see in the section introducing our innovative framework, this feature has significant consequences on the development and maintenance of the actual DSS.

6.2.3.2 DSS Development Phases

Sprague and Carlson (1982) propose a DSS development methodology popular in the DSS literature, that can be broken down into two broad parts: an "action plan", made of four phases described in Table 6.5, and the ROMC methodology, “a process-independent model for organizing and conducting systems analysis in DSS” (p. 102). The ROMC approach is made of four components: representations, operations, memory aids, and control mechanisms (hence the four-letter acronym). Table 6.6 describes these components with some examples.

Table 6.5. The DSS action plan

Phase	Description
I. Preliminary study and feasibility assessment	Survey the user base and assess user needs. Conduct pilot projects to ascertain general characteristics and the implications to DSS needs.
II. Development of the DSS environment	Form a DSS group, articulate its mission, and define its relationships with other organizational units. tablish a minimal set of tools and data and operationalize them.
III. Development of the initial specific DSS	Identify, analyze, and design the initial specific DSS with the users. Upgrade the tools and the data in response to needs that evolve from dealing with users.
IV. Development of subsequent Specific DSS	Assess needs for subsequent specific DSS. Develop subsequent DSS based on the initial one.

Table 6.6 The ROMC components

Component	Description	Examples
Representations	Conceptualizations used as methods of communication between the user and the DSS	Charts, tables, reports, input forms, equations
Operations	Activities necessary for the DSS to perform or facilitate the generation and delivery of the representations	Diagnosing and structuring problem; gathering, manipulating, and validating data; generate and assign risks to alternatives; simulate results of alternatives
Memory aids	Components intended to provide support to the use of the various identified representations and operations	Databases, workspaces, links, triggers alerting DSS users to perform specific operations, user profiles

Component	Description	Examples
Control mechanisms	Aids intended to help decision makers use representations, operations, and memory aids to synthesize a decision-making process based on their individual styles, skills, and knowledge	User interface aids, such as menus or function keys; help commands, tutorials; editing tools

Whereas the ROMC methodology leans more toward system engineering aspects (even though it does not specify how each of the four components is actually implemented), the phases of the action plan cannot be broken down into phases focusing on decision making and phases focusing on system engineering, because each phase attaches attention to both kinds of operations. This can be explained by the cardinal rule that the authors apply to the methodology: a DSS is to be driven by user needs. In other words, "a DSS effort should not be undertaken unless the need is apparent. Assessing the extent of need, however, requires some understanding and commitment to DSS in advance (another chicken-and-egg issue)" (p. 67).

Therefore, the authors include prototyping activities in the first phase already (in the form of pilot projects) and use the continuous and iterative assessment of what is called a "minimal set" of tools and data to gradually extend the project until it becomes a specific DSS. Nevertheless, this methodology remains high-level in nature and only supports the implementation phase of the DSS itself in a limited way.

6.2.3.3 Decision Support Engineering

The last methodology of this review, "Decision Support Engineering" (DSE), is proposed by Saxena (1991) as a "*comprehensive methodology based on a life cycle model of DSS development, which encompasses an engineering approach to DSS analysis and design*" (p. 99). Prototyping is also an important part of this methodology. The process is illustrated in Figure 6.4. The boxes identifying decision-making support steps and system-engineering steps have been added for the purpose of this review. The requirement engineering and DSS design steps overlap both categories, because some of their substages belong to decision-making support, while others belong to system engineering.

Much like Keen and Scott-Morton and others, Saxena is well aware of the strong and often turbulent relationship between the decision maker and the DSS builder: "whereas the decision maker's perception of 'support' opportunity provides a descriptive and contingency view of support, the DSS builder's perception is mainly technology-driven and prescriptive." Therefore, the author defines the DSS development process as a process of negotiation between the decision maker and the builder.

As illustrated in Figure 6.4, the decision support engineering methodology consists in five phases, each phase being broken down into a number of tasks. Trying to overcome the limitations of the traditional life-cycles models for DSS development, the DSE methodology proposes a separate phase for understanding the task structure before trying to model it. Much like Keen and Scott-Morton's design cycle, this approach is typical of a sequential process where software engineering tasks follow decision support tasks. The author makes an abundant use of feedback loops between the various phases of the methodology. However, he fails to describe how the builder can go back to a previous stage in the case of a goal failure at a later stage. This is a problem found in many DSS development methodologies: "*many papers simply state that an evolutionary approach was used without further elaboration*" (Arnott 1998).

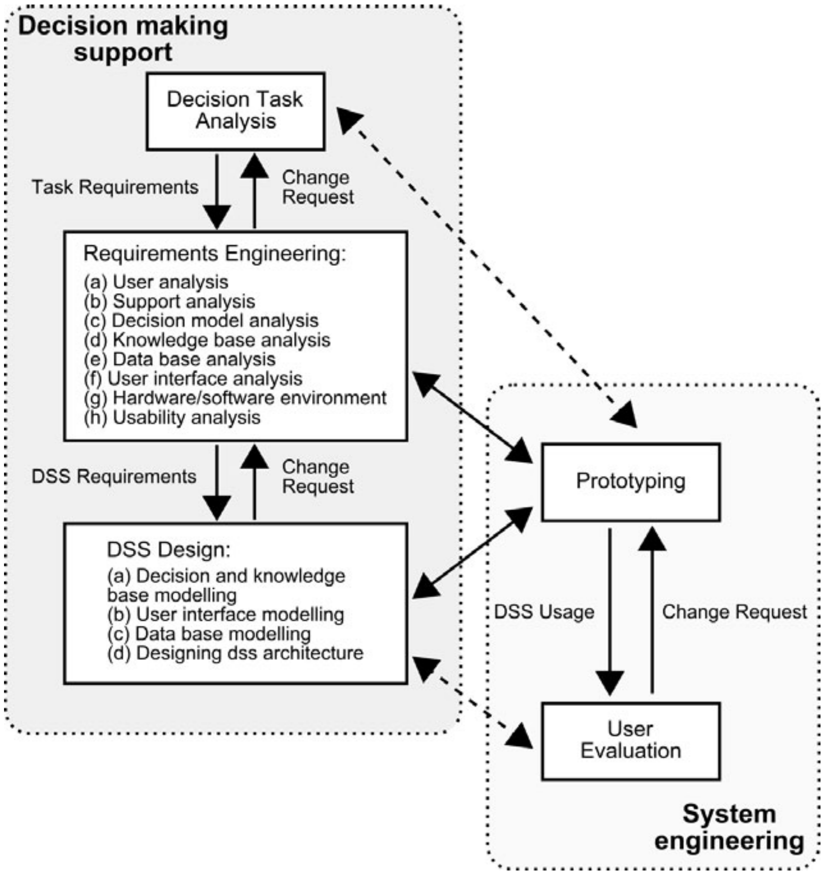


Figure 6.4. The Decision Support Engineering Methodology

6.3 A Tripartite Approach

It is no surprise that, over the years, the third, integrative form of DSS development processes gained in popularity inside the DSS community and finally prevailed over processes focusing solely on decision making or on system engineering. What is more interesting, though, is that no single development process predominates, as if all existing processes were recognized more as normative reference models than as usable descriptive processes.

We believe that a large part of this problem is due to two factors. On the one hand, the traditional organizational/technical dichotomy in DSS development failed to clearly position the design and development phases of the increasingly dominant knowledge-base and knowledge-engine components of the DSS. As explained in the review, these processes are often straddling decision-making and system-engineering phases, without any clear method.

On the other hand, almost all the development processes described in the previous sections, despite a sincere effort to foster an iterative design, keep making system-engineering aspects linearly dependent on the decision-making aspects. This fact is clearly illustrated in Figures 6.3 and 6.4, where the steps dealing with system engineering systematically follow the steps dealing with the decision support functionality. Yet the development processes of the purely technical components of the system and of the knowledge base of the DSS are very different in terms of development time, paradigms, tools, technical evolutions and the expertise required by the developer.

In this section, we propose an innovative approach taking these fundamental differences into consideration. The novelty of this tripartite approach lies in the clear and generic separation between the container of the DSS (responsible for the system engineering part) and the contents of the DSS (responsible for the knowledge base part). The interface layer between the container and the contents is managed by a component called the DSS kernel. This framework also improves the independent reuse of the container and the contents. As the underlying technologies (for example, programming languages on the container side and modeling systems or data base management systems on the contents side) evolve in very different ways, this clear separation can greatly increase productivity when developing specific DSS. Figure 6.5 proposes an integrated view of our proposition. This development approach replaces the traditional bipartite focus on decision-making support and system engineering with a tripartite focus on the knowledge base, knowledge engineering, and system engineering.

6.3.1 Container

The container is responsible for the system part of the DSS. Being decoupled from the contents of the DSS, its development can follow the traditional approaches described in the Section "Processes with a Focus on Software Engineering" (for example, the software development life cycle, prototyping, or even a controlled form of end-user development taking advantage of fast user interface design and personal exploration). The development of the container can also be supported by UML-based software modeling environment. A typical development cycle for a container lasts a few months, up to a couple of years. Due to the dynamic and rapidly changing nature of programming languages and software-engineering paradigms, containers are naturally subject to continuous, short-lived, and deep changes in the underlying technology.

Two system attributes of DSS are precustomization and customizability (Silver 1991). Precustomization is defined formally as "the degree to which, and the manner in which, at the time it is delivered to the user, some or all of the features of a DSS have already been tailored to the specific decision-making environment it is intended to support." Customizability is defined formally as "the degree to which, and the manner in which, a DSS empowers its users to specialize it as needed to fit the environment it supports." On the one hand, highly precustomized systems are attractive because they are already tailored to specific decision situations and decision makers, which is a central objective to many views of DSS (Keen and Scott Morton 1978). On the other hand, highly customizable systems are attractive because they can achieve customization while requiring less effort by the developer than would constructing a precustomized system. Customization is also necessary to continuously adapt the system to new use cases and evolving requirements[4].

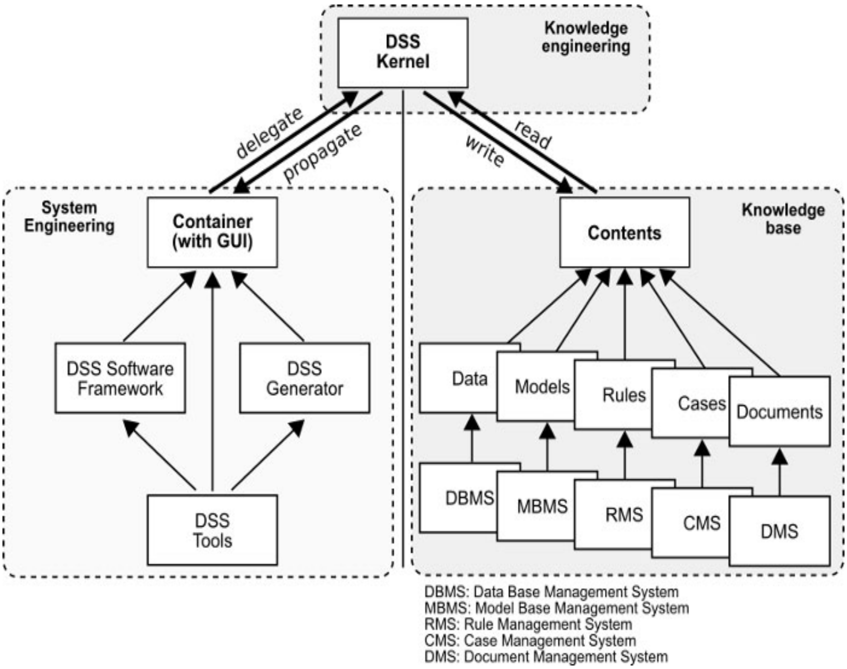


Figure 6.5. A tripartite approach

Given the very broad nature of DSS today, it is impossible to design a perfect DSS that would be both highly precustomized and highly customizable at the same time. However, as these two system attributes are well understood in software

4 The precustomization/customizability system attributes bear similarities to other design dimensions, such as the strong versus weak dimension used to analyze the influence of the system on the decision-making process (Moore and Chang 1983). Young (1989) defines the strong versus weak design dimension as "a relative measure of the extent to which the system leads or forces the user into a particular procedure (strong) or leaves it to the user to select or define what procedure to follow during any usage session (weak)" (p. 187).

engineering, it remains possible to apply them to the container of the DSS. To reach that goal, the basic functionalities of a container - ncluding those of the user interface - should not depend on any specific contents. Instead, the container should act as a kind of generic, empty shell offering low-level foundations for the contents. Then, specific requirements (covering the system architecture or the user interface) can be added by taking advantage of the customizable nature of the container. Gachet (2003) extends Sprague's classification of DSS development tools (Sprague 1980; Sprague and Carlson 1982) with a new component well adapted to design precustomized and customizable containers. This component is called a DSS software framework. As shown in Figure 6.5, it still remains possible to design containers using traditional DSS generators or DSS tools.

Nevertheless, separating the container from the contents raises many issues. To be successful, a container that should be reusable with different contents must definitely concentrate on generic system activities, such as user interface and input/output, user management, networking, security, cryptography, reporting, documentation, persistence, and system maintenance. The container should focus on these generic aspects, rather than on specific functionalities dealing with mathematical, relational, dimensional, document-, rule- or logic-based concepts. These aspects are under the responsibility of the contents of the DSS. For example, the authors created a Java- and Jini-based software framework for developing distributed cooperative DSS[5]. The software framework offers many of the generic functionalities listed above (Gachet 2004). It has been used to develop a distributed DSS for crisis management in the food supply sector, for the Swiss government (Gachet and Haettenschwiler 2003).

6.3.2 Contents

The contents are responsible for the knowledge base of the DSS and can be designed according to various paradigms (for example, model-based, rule-based, document-based, data-based, or knowledge-based) and processes (for example, the three processes presented in the Section “Processes with a Focus of Decision Support”). Their development is evolutionary and follows the declarative methods of knowledge engineering. A typical evolution cycle for the contents can last a few years, up to a few decades. Well-defined contents should be extensible and reusable in different containers.

Contents represent critical assets for an organization. Knowledge bases often formalize many years of expertise in the various domains of an organization. They are in continuous evolution and build a sort of corporate memory. To maximize its usefulness, this knowledge should be highly reusable, that is to say “available in the right forms to the right entities at the right times for the right costs” (Holsapple 2001). In our tripartite approach, contents are considered reusable if they can be coupled with different containers. Unlike containers, the technology underlying contents development is less subject to rapid changes. Thus, as container and contents do not evolve at the same pace, they should definitely be loosely coupled.

---

[5] http://www.dicodess.org

Otherwise, changes in the container technology could imply unnecessary changes in contents, and *vice versa*.

According to Marakas (1999), “[by] knowledge, we mean the rules, heuristics, boundaries, constraints, previous outcomes, and any other 'knowledge' that may have been programmed into the DSS by its designers or acquired by the DSS through repeated use.” A domain-specific knowledge base is mostly based on data, models, rules, cases, and documents, which are traditionally managed by database management systems (DBMS), model base management systems (MBMS), rule management system (RMS), case management systems (CMS), and document management systems (DMS).

Modeling languages have been around for more than thirty years. Prolog, for example, is a logical modeling language created by Alain Colmerauer around 1972. The original goal of Prolog was to provide a tool for linguists, enabling the expression of logic instead of carefully specified computer instructions. Thirty years later, Prolog is still used in many AI programs and other rule-based DSS and expert systems. The fascinating aspect of such languages is that ten or twenty years old knowledge bases written in Prolog can still be integrated in new containers written using modern programming languages, whereas the original containers are most of the time considered as legacy systems doomed to be replaced sooner or later.

Needless to say, Prolog is not the only modeling language able to produce long-lasting knowledge bases. JESS or CLIPS are two other examples of rule-based languages that can be used with knowledge-driven DSS. LPL (Huerlimann 1999), GAMS (Castillo 2002), or AMPL (Fourer *et al.* 2003) are powerful mathematical modeling languages that can be used with model-driven DSS. Finally, many OLAP engines can be used with data-driven DSS, and modern CMS or DMS can be used with document-driven DSS. For example, the DSS for crisis management mentioned at the end of the previous section uses the LPL modeling language to manage the knowledge base of the system.

6.3.3 The Interface Layer Between Contents and Container

At the highest level, the interface layer between the container and the contents is realized in a component referred to as the DSS kernel. This component includes both the underlying decision support model and the knowledge engine of the DSS. The decision support model defines guidelines to perform generic tasks such as decision task analysis, support analysis, and functional requirements analysis. These tasks define the overall architecture of the DSS and influence both the container and the contents. The knowledge engine is traditionally considered as the “brains” of the DSS. “The data and the models come together here to provide the user with a useful application that supports the decision context at hand” (Marakas 1999). This classical view of the knowledge engine is represented in Figure 6.5 by the read/write relationship between the contents and the kernel. In our innovative approach, we extend the role of this component by adding a delegate relationship between the container and the kernel. This is necessary if we want to decouple the container from the contents. We said in a previous section that the container should act as a kind of empty shell offering low-level foundations for the contents. To reach that goal, the container cannot deal directly with the contents (that is, the data, models, rules, or documents), hence the delegation of the operations accessing the contents to the kernel and, ultimately, to the knowledge engine. Obviously, the kernel needs to allow the container to communicate with it. At the program level, this can happen in the form of statically or dynamically linked libraries or, in the era of Web services, using a service-oriented architecture. In that sense, the container is completely decoupled from the contents.

Lower-level implementation of this interface layer can take several forms and the framework voluntarily avoids enforcing one solution over the others. For example, the interface layer between the container and the contents can be realized in components called decision support objects (Gachet and Haettenschwiler 2003). Decision support objects (DSO) can be informally described as reusable software envelopes placed by the container at the disposal of the contents. The natural object-oriented approach of DSO simplifies the task of the DSS user and increases their efficiency. As a well-defined object, a DSO fits perfectly in an object-oriented software paradigm on the container side. It can easily be exchanged in a distributed environment. As a decision support object, a DSO remains closely related to the specifics of a decision-making process. On the contents side, it encapsulates a business logic centered on decision support.

A container could offer a basic set of precustomized DSO for basic decision contexts (for example, DSO encapsulating dimensions, facts, assumptions, tasks, and exogenous decisions), yet should remain customizable in the sense that specific DSO could be added at a later time for a specific decision context. When loading contents, the kernel simply fills the empty DSO provided by the container with the appropriate data, model, rule, or document components. The container then lets the user organize the DSO according to the decision context and delegates all the decision-making operations (such as knowledge retrieval or alternative generation) to the knowledge engine inside the kernel. This concept is in line with modern approaches to DSS, that advocate a synergy between the decision support and knowledge-management processes of the organization (Carlsson and Turban 2002). On the one hand, decision-making processes driven by the container generate new knowledge (in the form of new DSOs), whose business logic is eventually stored in the contents by the knowledge engine (the write relationship in Figure 6.5). On the other hand, knowledge management activities influence the creation of new decision models to be adopted, which are eventually loaded by the knowledge engine from the contents (the read relationship in Figure 6.5) and propagated to the container in a new DSO (the propagate relationship in Figure 6.5). In the approach described by Gachet and Haettenschwiler (2003), DSO representing tasks precisely allow the DSS users to shape the decision space and, thus, the underlying decision process. First, decision assistants can clearly specify, manage, and organize their respective tasks in the container; then, model experts can prepare relevant models and procedures to fulfill the tasks in the contents. This interface provides two views: one for the decision assistant (using the container) and one for the model expert (defining the contents). The communication between users and DSS builders is formalized through these views and allows for future developments initiated by both sides. In that sense, the DSS kernel is the cornerstone of a bidirectional and loosely coupled communication between the container and the contents of the DSS, ensuring that the data, models, rules, documents, and the container remain flexible components.

6.4 DSS Development Processes Revisited

This section revisits two integrated methodologies presented in the review, according to the framework presented in the previous section. The methodologies in question are Saxena's decision support engineering process and Keen and Scott-Morton's design cycle.

6.4.1 Decision Support Engineering

We selected the decision support engineering methodology as a first example because its reorganization according to the tripartite approach is very obvious and illustrative. Figure 6.6 shows the revisited decision support engineering methodology and should be compared with the original process of Figure 6.4.

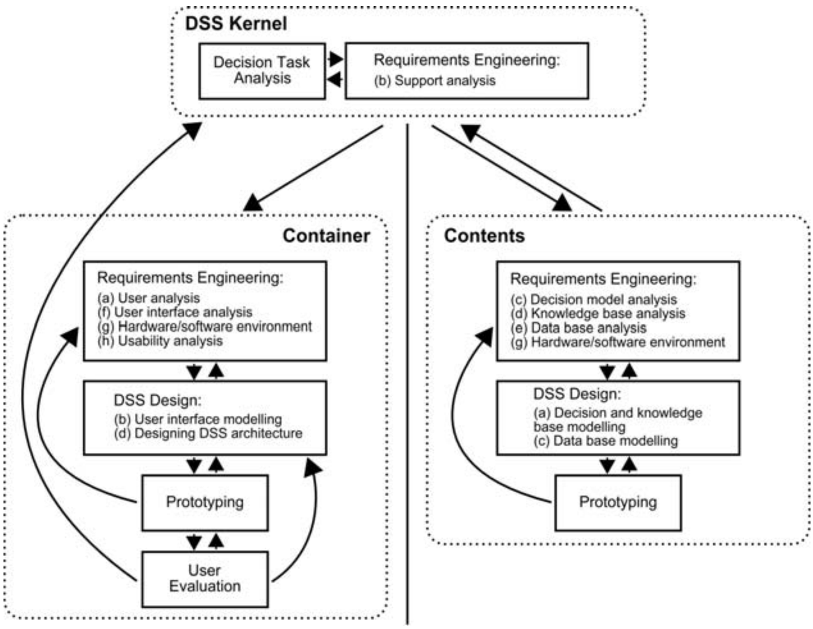


Figure 6.6. The "Decision Support Engineering" revisited according to the tripartite approach

The three components of the development framework (container, contents, and kernel) are clearly identified in Figure 6.6. As in the original methodology, the process starts with the decision task analysis. The next step of the original methodology, requirement engineering, breaks down into the three main components. The letters used in parenthesis to identify the subtasks have been kept as in the original process (Figure 6.4).

Support analysis, whose purpose is to gain knowledge about users' expectations and "provide a set of objectives of the DSS which define its support" (Saxena 1991, p. 102) clearly defines the operations that will be carried out by the DSS and that will impact the development of both the container and the contents. On the one hand, the contents is responsible for providing the data and models necessary for the operations. This responsibility is taken care of by the decision model analysis, knowledge-base analysis, data-base analysis, and the hardware/software environment related to the DBMS and MBMS subtasks of the requirement engineering step. On the other hand, the container is responsible for providing the basic system infrastructure and interface of the DSS. It forwards the user input to the knowledge engine and displays the reports and output propagated in return. User analysis, user interface analysis, usability analysis, and the hardware/software environment related to the user interface and the system part of the DSS belong to this part.

In the same way, the DSS design phase can also be divided between the container and the contents. User-interface modeling and the design of the general system architecture belong to the container. Decision and knowledge-base modeling, as well as data base modeling, belong to the contents part. Then, each part can be implemented independently and iteratively using prototypes. Discrepancies detected during the user evaluation in relation to objectives can lead to adjustments in the task and support analysis (in the knowledge engine), implemented as modifications in both the container and the contents. Note that the contents is usually evaluated by the user using the complete system (with the container side) and not in isolation.

An important difference between the original and revisited design cycles is that the revisited one explicitly takes into consideration independent design and prototyping cycles for both the container and the contents. As previously explained, each component needs its own development life cycle because it is noticeably different from the other. As a consequence, the development steps focusing on decision-making and those focusing on system engineering are not sequential on a global level (as in Figure 6.4), but only inside the individual container and contents components. It naturally follows that parallel design and implementation, as well as reuse, are greatly enhanced, since the container and the contents are not supposed to directly depend on each other. This approach allows for a better iterative design than the original, monolithic process.

6.4.2 The Design Cycle

The second revisited methodology is the design cycle. We chose this process because it is widely recognized in the DSS literature and it still influences many researchers and practitioners. This example helps reinforce the advantages that the tripartite approach brings with respect to existing processes. Figure 6.7 illustrates the revisited design cycle and should be compared with the original design cycle of Figure 6.3.

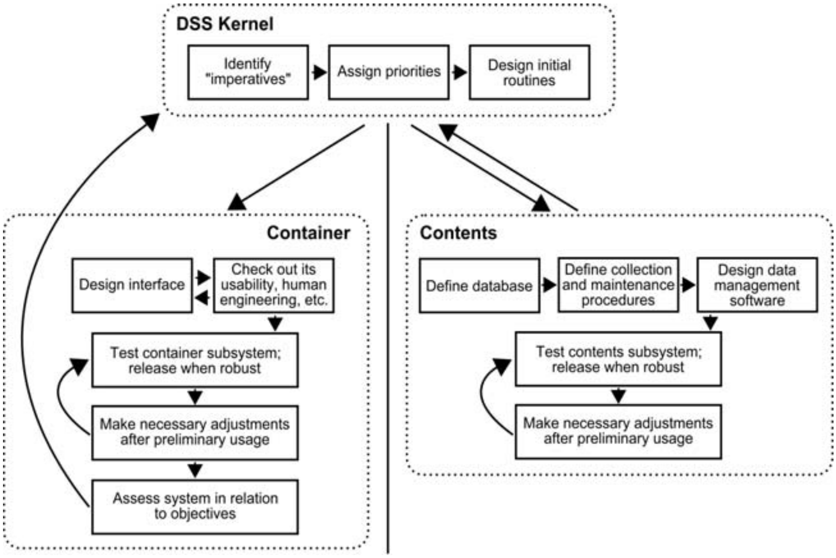


Figure 6.7. The design cycle revisited according to the tripartite approach

As in the original design cycle, the process starts with the identification of the functional requirements of the DSS (the "imperatives", or operations that the DSS should provide). The actual implementation of these functionalities impacts the two other components of the approach. On the one hand, the contents is responsible for providing the data necessary for the operations (Keen and Scott-Morton only dealt with data in their original design cycle, but models, rules, or documents could easily be added in the contents part) and the corresponding database-management systems. On the other hand, the container is responsible for providing the basic system infrastructure and interface of the DSS.

Discrepancies detected during the system assessment in relation to objectives can lead to adjustments in the imperatives defined by the knowledge engine, which will be implemented as modifications in both the container and the contents.

As in the revisited decision-engineering process, an important difference between the original and revisited design cycles is that the revisited one explicitly takes into consideration independent testing and adjustment cycles for both the container and the contents. As a consequence, the development steps focusing on decision making and those focusing on system engineering are not sequential on a global level (as in Figure 6.3), but only inside the individual container and contents components. The same advantages in terms of iterative development and reuse also apply to this revisited development process.

6.5 Concluding Remarks

It is fair to say that advances in knowledge management and artificial intelligence rejuvenated the field of decision support, bringing a new form of intelligence into

DSS. However, these advances also reshape the DSS development perspective – which used to deal mostly with organizational and technical issues – and highlight difficulties in adapting to modern knowledge management issues.

Based on a thorough analysis of existing DSS development processes and methodologies, this chapter explains that the development processes of the purely technical components of the system and of the knowledge base of the DSS are very different in terms of development time, paradigms, tools, technical evolutions and the expertise required by the developer. Consequently, it proposes a new framework for the design and development of DSS, whose cornerstone is the clear separation between the container of the DSS (responsible for the system part) and the contents of the DSS (responsible for the knowledge base). At the highest level, the interface layer between the container and the contents is realized in a component referred to as the DSS kernel. This component includes both the underlying decision support model and the knowledge engine of the DSS. The kernel manages the tasks influencing the development of both the container and the contents of the DSS.

This innovative approach still has a few limitations that should be dealt with in the future. In particular, it gives quite a bit of responsibility to the kernel component, which has to be able to communicate with both the container and the contents of the DSS. Another limitation is that the proposed approach does not consider the context of the target DSS, even though experience shows that the development of a large strategic DSS for a multinational company is very different from the development of a small-scale logistics DSS for a local company. In other words, no DSS development methodology can be considered as a "one-size-fits-all" solution, and our approach is no exception. Gachet and Sprague (2005) propose a partial solution to this problem.

Despite these limitations that merit special attention for future research, the proposed framework has important applications in that it can help DSS builders have a better understanding of the separate development processes of the container and the contents of the DSS, as well as of the links between the two components through the DSS kernel. As tripartite superstructure in which existing methodologies focusing on decision support or on system development can be reused, the framework presented in this chapter should help the field of DSS development leave the expansion stage to enter the control stage. Finally, it is our hope that this framework provides a vehicle for researchers and practitioners to develop better DSS systems and architectures.

Acknowledgements

This research is supported by Swiss NSF grant Nr. PBFR2-104340. The authors would like to thank Dr. Manuel Mora, the editorial committee, and anonymous reviewers for their valuable inputs and suggestions.

References

Alavi M. (1984) An assessment of the prototyping approach to information systems development. Communications of the ACM 27(6): 556-563.

Alavi M., Weiss IR (1985) Managing the risks Associated with end-user computing. Journal of Management Information Systems 2(3): 5-20.

Angehrn A, Jelassi T (1994) DSS research and practice in perspective. Decision Support Systems 12(4-5): 267-275.

Arinze B (1991) A contingency model of DSS development methodology. Journal of Management Information Systems 8(1): 149-166.

Arnott DR (1998) A framework for understanding decision support systems evolution. 9th Australasian Conference on Information Systems, Sydney, Australia: University of New South Wales.

Blanning RW (1979). The functions of a decision support system. Information and Management 2 (September): 71-96.

Bonczek RH, Holsapple CW, Whinston AB (1982). The evolution from MIS to DSS: Extension of data management to model management. Decision Support Systems. M. J. Ginzberg, W. R. Reitman and E. A. Stohr. Amsterdam, North-Holland: 61-78.

Burback RL (1997) Software Engineering Methodology: The WaterSluice, Database Group, University of Stanford. Retrieved March 15, 2005, from http://www-db.stanford.edu/~burback/water_sluice/sluice6.25.97/ws/watersluice.html.

Carlsson C, Turban E (2002). Decision Support System: Directions for the Next Decade. Decision Support Systems 33(2): 105-110.

Castillo E (2002) Building and solving mathematical programming models in engineering. New York, Wiley.

Courbon JC, Drageof J, Tomasi J (1979) L'approche évolutive, Informatique et Gestion 103(Janvier-Février).

Courbon JC, Grajew J, Tolovi J (1980).Design and Implementation of Decision Supporting Systems by an Evolutive Approach, Unpublished working paper.

Eom SB (1998) An overview of contributions to the decision support systems area from artificial intelligence. AIS Conference, Baltimore, MA, USA.

Forgionne GA, Mora M. Cervantes R, Gelman O (2002) I-DMSS: a conceptual architecture for next generation of DMSS in the Internet age. International Conference on Decision-making and Decision Support in the Internet Age (DSIAge02), Cork, Ireland: 154-165.

Fourer RD, Gay M, Kernighan BW (2003) AMPL: a modeling language for mathematical programming. Pacific Grove, CA, Thomson/Brooks/Cole.

Gachet A (2003) Software frameworks for developing decision support systems: a new component in the classification of DSS development tools. Journal of Decision Systems 12(3/4): 271-281.

Gachet A. (2004) Building Model-Driven Decision Support Systems with Dicodess. Zurich, vdf Hochschulvlg.

Gachet A, Haettenschwiler P (2003) A Jini-based software framework for developing distributed cooperative decision support systems, Software - Practice and Experience 33(3): 221-258.

Gachet A, Sprague R (2005) A context-based approach to the development of decision support systems, International workshop on Context Modeling and Decision Support, Paris, France.

Gibson C, Nolan R (1974). Managing the four stages of EDP growth. MoIS: Management of Information Systems. P. Gray, W. King, E. McLean and H. Watson. Forth Worth, TX, The Dryden Press.

Holsapple CW, Whinston AB (1996) Decision support systems : a knowledge-based approach. Minneapolis/St. Paul, West Pub. Co.

Holsapple CW (2001) Knowledge Management Support of Decision-making, Decision Support Systems 31(1): 1-3.
Huerlimann T (1999) Mathematical Modeling and Optimization, An Essay for the Design of Computer-Based Modeling Tools. Dordrecht, Kluwer Academic Publishers.
Hurst, EG, Ness DN, Gambino TJ, Johnson TH (1983) Growing DSS: a flexible, evolutionary approach. Building Decision Support Systems. J. L. Bennett. Reading, MA, Addison-Wesley: 111-132.
Keen, PGW. (1980a) Adaptive design for decision support systems, Database 12(1-2): 15-25.
Keen P. G. W. (1980b). Decision support systems: a research perspective. Decision support systems : issues and challenges. G. Fick and R. H. Sprague. Oxford ; New York, Pergamon Press.
Keen PGW, Scott-Morton MS (1978) Decision support systems : an organizational perspective. Reading, Mass., Addison-Wesley Pub. Co.
Kreie J, Cronan TP, Pendley J, Renwick JS (2000) Applications development by end-users: can quality be improved ?, Decision Support Systems 29(2): 143-152.
Marakas GM (1999) Decision support systems in the twenty-first century. Upper Saddle River, N.J., Prentice Hall.
Martin MP (1982). Determining information requirements for DSS., Journal of Systems Management (December): 14-21.
McNeese C, Hayes O, Bonneau S (2004) Shining the Light on Shadow Staff, CIO.com. Retrieved February 28, 2005, from http://www2.cio.com/consultant/report2085.html.
Moore JH, Chang MG (1983). Meta-Design Considerations in Building DSS. Building Decision Support Systems. J. L. Bennett. Reading, MA, Addison-Wesley: 173-204.
Nolan R (1979). Managing the crises in data processing., Harvard Business Review 57(2): 115-126.
Paulk MC (1998) The Capability Maturity Model for Software, Version 1.1, Software Engineering Institute, Carnegie Mellon University. Retrieved March 15, 2005, from ftp://ftp.sei.cmu.edu/pub/cmm/cmm-over.pdf.
Power DJ (2002) Decision support systems : concepts and resources for managers. Westport, Conn., Quorum Books.
Raden R (2005) Shedding Light on Shadow IT: Is Excel Running Your Business?, DSSResources.com. Retrieved February 26, 2005, from.
Sage AP (1991) Decision support systems engineering. New York, Wiley.
Saxena KBC (1991). Decision support engineering: a DSS development methodology. 24th Annual Hawaii International Conference on System Sciences (HICSS'91), Los Alamitos, CA, IEEE Computer Society Press.
Saxena KBC (1992) DSS Development Methodologies: A Comparative Review. 25th Annual Hawaii International Conference on System Sciences (HICSS'92), Los Alamitos, CA, IEEE Computer Society Press.
Scott-Morton MS (1971) Management decision systems. Boston, Massachusetts, School of Business Administration, Harvard University.
Silver M (1991) Systems that support decision makers : description and analysis. Chichester ; New York, Wiley.
Sprague RH (1980) A framework for the development of decision support systems, MIS Quarterly 4(4): 1-26.
Sprague RH, Carlson ED (1982) Building effective decision support systems. Englewood Cliffs, N.J., Prentice-Hall.
Stabell CB (1983) A Decision-Oriented Approach to Building DSS. Building Decision Support Systems. J. L. Bennett. Reading, MA, Addison-Wesley: 221-260.
Young, L. F. (1989). Decision support and idea processing systems. Dubuque, Iowa, W.C. Brown.
Tiwana A (2002) The knowledge management toolkit : orchestrating IT, strategy, and knowledge platforms. Upper Saddle River, N.J., Prentice Hall.

7

Explanatory Power of Intelligent Systems

Robbie T. Nakatsu

College of Business Administration, Department of Computer Information Systems,
Loyola Marymount University, 1 LMU Drive, Los Angeles, CA 90045, USA
e-mail:makatsu@lmu.edu

This chapter provides a framework for understanding the explanatory power of intelligent systems. It looks at content-based enhancements, drawn primarily from the expert systems literature, interface-based enhancements, and the appropriate selection of an advisory strategy. Such enhancements contribute to explanatory power by increasing system transparency and flexibility, and lead to outcomes such as better decision-making and problem-solving performance. Three illustrative examples demonstrate each type of enhancement of explanatory power. In the first case, a graphical hierarchy of an expert-system knowledge base is illustrated. In the second case, the use of restrictive vs. nonrestrictive advisory strategies is discussed. Finally, deep explanations that provide a deeper understanding of a domain of expertise are described.

7.1 Introduction

Intelligent systems provide advice to the end user to assist in decision-making and problem-solving (Carroll and McKendree,1987). One criticism levelled against these systems is that they are rigid dialogs that hard to understand (see *e. g.*, Hayes-Roth and Jacobstein 1994, Franklin 1997). In Expert Systems, for example, it is hard to understand how a given set of inputs produced a recommendation by the system. To the end-user, the expert system's reasoning process is a "black box". To address this problem, this chapter argues that such systems require greater explanatory power in order for users to accept their recommendations and have more confidence in the decision support that they provide.

Explanatory power refers to the ability of an intelligent system to explain its actions (Nakatsu 2001). Two related characteristics are relevant to understanding explanatory power: **transparency**, or the ability to see the underlying mechanism of the system so that it is not merely a black box; and **flexibility**, or the ability of the interface to adapt to a wide variety of end-user interactions, so it is not merely a rigid dialog, but an open-ended interaction that allows the end user to explore and understand the system more fully. While transparency of the system is a quality related to the informational content of the system itself, flexibility is more related to the nature of the end-user interaction with the system. The distinction is a subtle one, for it is easy to confuse the two qualities. More flexibility in the user interface can lead to more transparency—having a more open-ended interaction can enable an end-user to seek out more ways to better understand the system; by the same token, having a more restrictive interface can impair the end-user's ability to seek out more transparency in the system, even if it does already exist (Schank 1986). Hence, this research views flexibility as a separate quality of a user interface that exists independently of interface transparency. The explanatory power construct, as defined in this chapter, integrates and captures both transparency of the user interface and flexibility of the end-user interaction.

Making the distinction between informational qualities vs. interaction-related qualities of user interfaces is an important first step in developing a framework for explanatory power. Our goal is to describe and better understand the multi-dimensional nature of explanatory power—specifically, to investigate some of the determining factors of explanatory power. By doing so, we will be in a better position to offer design guidelines, which can be used by system designers to enhance a system's effectiveness.

The research framework looks at three types of enhancements to explanatory power. **Content-based enhancements** focus on augmenting the actual informational content of an intelligent system. We review the different types of explanations that are possible, drawing primarily from the expert systems literature. However, offering more content alone may not offer any real gains in explanatory power, because the end user may simply not bother to utilize the explanations. Hence, this research also looks at ways to enhance the interactive experience of the end user. Two additional determinants of explanatory power, related to enhancing the interactive experience are considered. **Interface-based enhancements** relate to the interface design choices that systems designers make to increase the effectiveness of intelligent systems. A number of possibilities are suggested that may increase the flexibility of the user interface, and as a result, its overall usability and usefulness. Still, the interface characteristics, in and of themselves, may not result in improved problem-solving performance. Hence, a third type of enhancement is the appropriate selection of an **advisory strategy**, or the manner in which the explanation is delivered to the end user.

7.2 Content-based Enhancementes

How can we improve the informational content of intelligent systems so that their internal mechanism is more visible for inspection (*i. e.*, the transparency of the system is enhanced)? This section will address content-based enhancements to explanatory power. The research involved in content-based enhancements has a long and well-documented history in the artificial intelligence (AI) and expert systems literatures, which this section will review. From this overview, a classification of explanation types by content is provided.

Computer-generated explanations have long been associated with expert systems use. Explanations use in expert systems begins with MYCIN, developed by Edward Shortliffe at Stanford Medical School in the 1970s for the diagnosis and treatment of bacterial infections (Buchanan and Shortliffe, 1985). MYCIN is considered one of the classic expert systems, certainly the most widely cited, and it has introduced several features that have become standards in expert systems technology: rule-based knowledge representation, probabilistic rules to capture uncertainty, the backward chaining method, explanation, and a user-friendly interface (Turban, 1995). To aid with system debugging, Shortliffe added a RULE command that, when requested, asked MYCIN which rule was currently being used. The rule was displayed in LISP, but later was displayed in English to make this simple explanation more user-friendly (Buchanan and Shortliffe 1985). Later, this RULE command was changed to WHY to enable "the user to examine the entire reasoning chain upward to the topmost goal by asking WHY several times in succession." (*idem*. p. 333). The HOW explanation was also developed that enabled the user to descend the branches of the reasoning network. Today, WHY is commonly used by the user to ask why a certain type of input is needed by the system (typically the rule requiring the input is displayed). The typical HOW question is posed by users when they would like to know how a certain recommendation or conclusion was reached (typically the entire rule trace of the reasoning process is given).

Why does the user need an explanations facility? Buchanan and Shortcliffe (1985) offer several reasons. First, both the system builder and the end user need to understand the knowledge base in order to maintain it and use it effectively. Secondly, systems builders can use explanations for debugging purposes. Thirdly, explanations serve an educational function. Users who feel they learn something about the knowledge base are more likely to feel more comfortable with such a system. Fourthly, explanations can help to convince users that the conclusions the Expert System reaches are reasonable, and lead to their acceptance.

7.2.1 Limitations of Rule Traces

Explanations based on rule traces are clearly limited in terms of providing explanations that are instructive, inclusive, and easy to use. For one, the how-why paradigm of explanations use described above offers a very limited form of explanation. Other types of questions might need to be asked, especially if the end-user feels unsure of the system's advice. For example, "what-if" questions might enable users to explore the effect of changing assumptions in the rule-base, or to perform sensitivity analysis on input variables. "Why-not?" questions or "Why did you not conclude that <such and such> is true?" are frequently asked when probing real human experts (Ellis, 1989), yet this capability is a relatively difficult one to develop in an expert system.

Probably more problematic is that the "why" and "how" questions are based on rule traces, which an end user is likely to have difficulty in comprehending. One solution is to replace rule traces produced directly from computer code, with canned text that is easier for the end user to understand. However, the replacement of computer-generated explanations with canned text explanations comes at a stiff price: user questions must be anticipated in advance and it is unlikely that all such questions will be thought of ahead of time (Moffit 1994). Maintenance of such canned text explanations (keeping them in sync with an ever-changing rule base) could also create problems further down the line. Moreover, by using canned text explanations, the system has no conceptual model of what it is saying so that it is not possible to develop more advanced types of explanations, such as providing explanations at different levels of abstraction to the end-user (Swartout 1983).

Another problem with rule traces is that they sometimes provide too much detail, which an end user is simply not interested in seeing. As Swartout (1983) astutely observes in discussing the MYCIN rule traces: "Parts of the program [*i. e.* rule traces] appear mainly because we are implementing an algorithm on a computer. If these steps are described by physicians, they are likely to be uninteresting and potentially confusing." (p. 312). There is often too much algorithmic detail in a rule trace that an end user cannot understand, or may not care to understand. An effective explanation must be pitched at a higher level, so that unnecessary details are left out.

Several researchers have also commented on the **opacity** of rules, or a rule's inability to make visible to the end-user the underlying reasoning process. Clancey (1983), for one, has noted that rules typically do not contain justifications or fail to shed light on underlying causal processes. This may be due, in part, to the way that expert knowledge is compiled: "rules are 'compiled' in the sense that they are optimizations that leave out unnecessary steps—evolved patterns of reasoning that cope with the demands of ordinary problems." (p. 225) However, these intermediate steps frequently need to be explained to end users who may not understand how an expert system reached a conclusion.

Clancey (1983) has also pointed out how strategic knowledge can be hidden in the premises of rules. He defines strategic knowledge as "an approach for solving a problem, a plan for ordering methods so that a goal is reached" (p. 233). For example, these rules, by the simple ordering of the premises, dictate to the inference engine which conditions should be checked before the others. Such strategic knowledge is effectively lost to the end-user requesting a rule trace.

7.2.2 More Sophisticated Explanations

The preceding discussion suggests a number of improvements to traditional rule traces, some of which have already been explored by a number of researchers. Obviously, an explanation facility cannot do everything, and part of the problem of good explanation design lies in understanding what types of explanatory capabilities can be feasibly developed, for a given task domain and for a given class of users. The following discussion considers some extensions to traditional rule traces.

In NEOMYCIN, an offspring of MYCIN, Clancey and Letsinger (1981) (see also Clancey 1983, Hasling *et al.* 1984), consider providing explanations that capture the overall approach used by the system to solve a problem—that is, the underlying strategic knowledge. One suggestion made to this effect is to capture strategic knowledge in metarules. These metarules provide the high-level knowledge for controlling the use of rules. An example of a meta-rule from NEOMYCIN is given below:

IF (1) the infection is pelvic-abscess, and
(2) there are rules that mention in their premise enterobacteriaceae, and
(3) there are rules that mention in their premise gram-pos-rods,

THEN there is suggestive evidence (.4) that the former should be done before the latter

The firing of the above rule will cause one goal (2) to be pursued before another goal (3). Such metarules make the expert system's problem-solving strategy more explicit and therefore, potentially more visible to the end-user requesting an explanation.

In XPLAIN, Swartout (1983) suggests capturing strategic knowledge as a tree of goals, called a refinement structure. "Refining" a goal means turning it into more specific subgoals. Hence, the top of the tree is a very abstract, high-level goal, and the lower levels of the tree represent less abstract steps needed to implement this goal. Eventually, the level of the system primitives (*i. e.* built-in system operations) is reached.

Aikens (1983) identifies the problem of the user being unable to follow a line of reasoning under traditional explanation facilities. For example, MYCIN is unable to deal with more than one rule at a time, so that a line of reasoning is provided as a result of a user requesting a succession of "WHY" explanations. To make the line of reasoning more visible, Southwick (1988), for one, has advocated a hierarchy of landmarks or topics to guide the end-user. He defines a topic as "a logical and conceptual entity in the knowledge base of an Expert System." Such topics serve as landmarks or anchoring points in the knowledge base. These topics, ideally, should have some intuitional appeal for an end-user so that a system can use them as convenient explanatory segments. In a similar vein, Mockler (1989)] utilizes dependency diagrams to graphically model a knowledge-based system. These diagrams show knowledge segments and their interrelationships, in a hierarchical fashion. They provide an overall summary view of the knowledge base. Such a tree structure could help end users to better comprehend a knowledge base, unlike a flat set of rules.

Another criticism leveled against traditional explanation facilities is that they are incapable of justifying their actions, or providing an underlying reason for a system action. One solution to this lack of justification are model-based explanations that justify system actions and results by linking them to a deep causal model of the domain (Southwick 1991). Such deep explanations are believed to give end-users an understanding of the underlying reasons for a recommendation. Swartout (1983) identifies two types of deep knowledge that might be provided as explanations: the **domain model** is descriptive knowledge about the domain and consists of such things as taxonomic knowledge and causal relationships; the **domain principles** are prescriptive knowledge and consist of such things as methods and heuristics used for problem solving.

Wallis and Shortliffe (1985) advocate the development of causal networks in order to create better explanations for medical consultation systems. The causal network, they contend, can serve as an integral part of the reasoning system, and can be used to guide the generation of customized deep explanations. For example, we might assume the causal chain in a system to be of the form t1 → t2 → t3 with each element assigned a measure of complexity. Suppose further that t2 is deemed to be too complex by a novice user. The system can tailor such explanations so that more fine-grained (and more complex) elements in the chain of causality are hidden from such users; in this case, t1 → t3 only is revealed to the user.

Model-based reasoning techniques may also be employed to create a deeper model of a domain of discourse. These techniques solve problems by analyzing the structure and function of a system, as described by a symbolic model (Kunz, 1987). Unlike rule-based expert systems, which reason from “canned” rule-based associations (IF-THEN rules), systems employing model-based reasoning contain a model simulating the structure and function of a system. Model-based reasoning systems, in large part, came into being to address the void created by systems employing only rule-based reasoning and other limited forms of reasoning (*e. g.* flowcharts, fault dictionaries, decision trees, bayesian probability theory). Much of the work done in this area has focused on fault diagnosis and troubleshooting, in particular, for devices such as electronic circuits. Model-based diagnosis, as Davis and Hamscher (1988) note, starts with an observed malfunction of a device, and works backward to determine which underlying component(s) might be the cause.

Research in model-based reasoning systems has not been limited to physical devices. Patil (1988) explore the Artificial Intelligence techniques used for diagnosing diseases in medicine. Of particular relevance to model-based reasoning is their experimental program called ABEL. ABEL’s knowledge base attempts to capture the underlying causal mechanisms of disease processes. ABEL views diagnosis as a process of building detailed causal models that explain a patient’s illness.

7.2.3 A Classification of Explanation Types: A Summary of Content-based Enhancements

This section summarizes the three types of explanations that contribute to the overall explanatory power of an intelligent interface: rule traces, strategic knowledge, and deep justifications. This classification is similar to taxonomies developed by other researchers (see *e. g.* Gregor and Benbasat 1999, Chandrasekeran *et al.* 1988, Southwick 1991). Under each type, points about how the explanatory power of the user interface may be increased are provided.

Rule traces:

- Different types of questions are allowed such as “why-not?” and “what-if?”.
- Rules are displayed in a natural language as opposed to computer-generated code.
- Rules traces are displayed at the right level of detail.

Strategic knowledge:

- The problem-solving strategies are made explicit to the end user (or are available upon request)
- The overall line-of-reasoning is made visible to the end user.
- The strategic knowledge is appropriately structured for the end user (*e. g.*, a tree of goals, or a tree of topics).

Deep Justifications:

- An explanation is tied to an underlying domain model that provides structural knowledge (*e. g.* causal relationships about the domain) and taxonomic knowledge about the domain.
- An explanation is tied to underlying domain principles that provide knowledge about methods and heuristics used to problem solving.

7.3 Interface-based Enhancements

Whereas content-based enhancements are concerned with enhancing the actual explanations themselves, interface-based enhancements focus on designing the user interface to foster transparency and flexibility in the system. The trend toward high-powered PCs and workstations during the 1980s and 1990s has led to the user interface taking on a more central role in the development of intelligent systems. In addition, the need to support the user's cognitive task has also necessitated that interface design considerations take on a more prominent role (Hayes-Roth and Jacobstein 1994, p. 40, Stelzner and Williams 1988). Indeed, most of the current and most powerful expert system shells on the marketplace contain powerful tools to design object-oriented, graphical user interfaces (*e. g.* Gensym Corporation 1997).

Still another reason for the importance of the user interface is the increasing size and complexity of systems. Large-scale, industrial strength systems in organizations require that the user interface manage complexity well. This means that such interfaces must enable end users to browse quickly through large amounts of information and to obtain multiple views of the same knowledge, in order to support the varying task requirements of the different users of the organization.

7.3.1 Characteristics of the User Interface

The careful selection of features of the user interface may also enhance the explanatory power of a system. The focus in this discussion will be on supporting the end user—providing him or her with the capabilities to interact with the system in a variety of ways, as well as supporting the management of cognitive complexity. Stelzner and Williams (1988) identify five major requirements of the expert system user interface, which will provide a framework for the discussion: 1) the natural idiom; 2) immediate feedback; 3) recoverability; 4) granularity; and 5) multiple interfaces to the same knowledge.

The natural idiom. This refers to the ability of the interface to represent the end-user's domain so that it maps as closely as possible to an end-user's mental model. Stelzner and Williams argue that the central metaphor of the interface should be that of the modeled world itself: instead of describing the domain (using a text-based conversational dialog, for example), the end user should perform actions directly on a graphical model of the domain. The end result is that the user interface is more natural and easier for an end-user to learn and use.

Immediate feedback. This refers to the ability of the interface to offer feedback to the user based on his or her actions. This quality supports the feeling of acting directly on the objects of the domain model, and removes the perception of the computer acting as an intermediary (*idem*). Animation is one technique that might be used to endow the system with immediate feedback. Stelzner and Williams provide the example of a knowledge-based simulation of a factory, in which the graphical user interface contains a detailed layout of the factory to aid engineers in the determination of what operating strategy is best. Animation of basket movement through the factory allows the engineer to quickly identify bottlenecks and underutilized resources in the factory.

Recoverability. This refers to the ability of the interface to allow the end-user to back out of changes made to the system. End users may wish to test out the effects of different changes on a system, and may desire an easy way to back out of these changes. Such a capability will encourage an end user to explore and experiment with a system's capabilities. There are many possible ways of building recoverability into an interface.

An interface endowed with a high degree of recoverability may include undo facilities, history lists of the most recent actions performed, hypertext links that enable the end user to jump back to the relevant portion of the problem-solving situation, and graphical user interfaces that enable end-users to select objects of the domain and easily modify their attribute values.

Granularity. This refers to the ability of the interface to display different levels of detail, depending on the situation the user is currently in. Especially in complex systems composed of multiple components, this capability is crucial to addressing the cognitive limitations of an end user. Such an end-user can become overwhelmed by the enormous amount of information required to understand the system. An interface that supports granularity may be one that enables the creation of multileveled, hierarchic descriptions of a domain. An end-user can drill down the branches of the hierarchy to obtain more detail, or go up the branches to obtain a higher-level view of the domain. An interface endowed with granularity would enable such a user to change the level of granularity frequently and with ease during a user session.

Multiple interfaces to the same knowledge. Given that different tasks and different users may have different requirements for utilizing knowledge, a system that enables the development of multiple interfaces to the same knowledge would permit greater flexibility. For example, in a real-time process control system, it might be critical for an operator to receive alerts whenever there is a malfunction in one of the components of the system. One interface might employ the use of multimedia, say the use of sound, to alert the user to a potentially dangerous situation. Another user of the system (say the supervisor) may not need these alerts, but would require hourly summaries of the outputs of the system, and the detection of underutilized resources in the system. A different interface and view of the system is obviously required.

7.4 Advisory Strategies

The third type of enhancement to explanatory power are the strategies employed to deliver the advice to the end user. Once explanatory content has been created, and the appropriate interface type and interface features selected, a designer of an intelligent interface may also utilize an appropriate advisory strategy to enhance the system's effectiveness. Selection of an inappropriate strategy may result in suboptimal usage of the explanations that the system offers: end users may not bother to request the explanations at all, or the explanations, when requested, may not be usable or useful at a given point during a user's interaction with the system. This section will consider different types of advisory strategies, and when and how they may be employed to enhance a system's effectiveness.

7.4.1 Types of Strategies

What are the different ways in which advice can be delivered to the end user ? A designer of an intelligent system may consider a variety of strategies that address the following questions: At what point during a consultation should advice be presented? Should advice be user invoked or automatically provided by the system? How much advice should be given? Should the system allow for "special" modes of operation that protect the system from unintended consequences?

The **timing of the advice** addresses the question of when the advice should be delivered. Dhaliwal and Benbasat (1996) distinguish between **feedforward** versus **feedback** advice, using the cognitive feedback paradigm (Todd and Hammond 1965). Under this paradigm, there are three differences between feedforward and feedback: temporal order, cues focused upon, and case specificity. In terms of temporal order, feedforward is always presented prior to task completion, while feedback is presented after task completion. In terms of cues focused upon, feedforward focuses on input cues, whereas feedback is advice related to outcomes. Finally, in terms of case specificity, feedback is case specific because it provides specific advice regarding the outcome (or recommendations) that the expert system makes, while feedforward tends to be more generic to the task at hand. Some researchers have chosen to think of feedforward as non-case-specific training provided prior to task performance. Case specificity is more a content issue (*i. e.* content-based enhancement) than an issue of timing, but it is useful nonetheless to point out how the timing of the advice can (and should) affect its content.

The **provision mechanism** is another type of strategy that can affect explanations usage. Two types of provision mechanisms are considered. **User-invoked** explanations are explicitly requested by the user, whereas **automatic** explanations are automatically provided as determined by the system (Gregor and Benbasat, 1999). This distinction corresponds to the active vs. passive distinction, which Fischer *et al.* (1985), employ in their research on help systems: active help systems interrupt the user's actions, while passive help systems wait until the user explicitly requests advice.

Moffit (1994, 1989) conducted an experiment on the effectiveness of user-invoked vs. automatic explanations provision. She called the automatic explanations "embedded-text" explanations, since these explanations were embedded within the interface dialogue, and hence, the end-user would automatically see them. Her experimental evaluation sought to discover which explanations provision mechanism would enhance learning the most when using a production-oriented scheduling expert system. Subjects were randomly assigned to one of four treatments: (1) no explanation; (2) user-invoked, rule-trace facility; (3) user-invoked, canned text facility; and (4) embedded text (automatic provision).

Both declarative and procedural knowledge were tested and measured. The embedded text treatment appeared to offer the greatest advantage in terms of learning. This result led Moffit to conclude that the more difficult it was to access the explanations (*i. e.* user invoked), the more the subjects perceived the expert system as a separate computerized tool—as opposed to a natural part of the human-computer interaction—and they, therefore, became less aware of its informational value.

Controlling the **amount of advice** is another way that a system designer may wish to affect explanations use. Having more advice is generally considered a good thing, and the more the better. However, there is a body of research that suggests just the opposite (see Carroll and McKendree (1987) for a summary): having advice could actually distract a user whose goal is something other than learning. Proponents of user discovery of the system, for example, argue that advice should be provided only when the user explicitly requests it, countering the design recommendation that Moffit's study seems to suggest—that explanations should be embedded within the dialog. As Carroll and McKendree (1987) observe, "*the discovery approach takes advantage of opportunistic learning, that is, making the most of each unique personal experience*" (p. 23). Many researchers seem to be in agreement with the discovery learning approach. Brown et al (1982) argue that providing large amounts of advice was often quite harmful, and that it is often better to leave the user alone, especially if the problem-solving task is small. They also suggest that no advice be provided at all if the user gets too far off track, the implication being that it is unclear what sort of advice can help a user in such a situation.

Finally, the use of **special modes** of system operation is considered another class of advisory strategies. Carroll and McKendree (1987) distinguish between two special modes of operation. **Control blocking** means that a portion, or subset, of the system's functions is made inaccessible to the user to prevent their accidental usage. One example of this type of strategy is to use a training wheels approach, in which a portion of the system is rendered inaccessible to novice users, who often access advanced functions by mistake, and then become distracted and confused by the consequences (*idem*). Carroll and Carrithers (1984) showed that such an approach could lead to more efficient learning of a word-processing application. Another type of special mode is a **protected mode** in which a user action is protected from harmful consequences. One type of protected mode is a *reconnoiter mode* in which the actions of system commands are simulated without actually affecting the system's data (Jagodzinski 1983). A user of this system can switch to reconnoiter mode and try different things out, without fear of destroying or damaging system data, and then return to normal mode. Such a mode of system operation can encourage the user to more fully explore a system.

7.4.2 A Summary of Advisory Strategies

This section summarizes the preceding discussion on advisory strategies. Selection of an appropriate advisory strategy can mean the difference between effective usage of explanations or ignoring the explanations altogether. This listing is by no means an exhaustive enumeration of the possiblities.

Timing of the advice:
- Appropriate use of feedforward and feedback should be made in the delivery of explanatory content.
- Feedback should be provided immediately after a system error has occurred (not delayed).

Provision mechanism:
- Automatic explanations should be provided if the system designer wishes the end user to use them since the cognitive effort required to use them is low.
- User-invoked explanations should be provided if the system designer does not want to force explanations on the end user; rather they are requested at the end-user's discretion.

Amount of advice:
- Too little advice means the interface lacks explanatory content.
- Excessive amounts of advice may hamper discovery learning: too much advice may be distracting.

Special modes:
- Control blocking may be used to limit an end-user's access to system functions that may be confusing and distracting to use.
- Protected modes may be implemented to promote user discovery and creative uses of information systems.

7.5 Three Illustrative Examples

Three examples of enhancing explanatory power are provided below. In the first example, a graphical user interface, which models the knowledge base of an Expert System, describes an interface-based enhancement. In the second case, an illustration of two advisory strategies, and how they might be employed for a problem-solving task is provided. In the third case, deep explanations that provide underlying domain principles illustrates a content-based enhancement.

7.5.1 Hierarchic Models of Intelligent Systems (Interface-based Enhancement)

Expert-Strategy (Nakatsu and Benbasat 2003) utilizes hierarchic models that are intended to partition an Expert System's knowledge base into more manageable chunks. These hierarchic models are meant to be interactive and allow the end-user to inspect parts of the hierarchy to better understand how the intelligent system reached a certain conclusion.

Figure 7.1 provides a screen shot of a hierarchic model used by Expert-Strategy. In this application, a transportation broker makes a determination concerning what type of transportation mode (air, trucking, rail, small package service) to use for a client's shipment. The topmost node of the hierarchy, Transportation Mode, represents the system recommendation. Several input variables are entered at the bottom of the hierarchy: shipment weight, weight of one item, value of one item, fragility rating, shipping distance, desired transit time, product perishability, and other factors. These inputs are directly entered onto the leaf nodes of the hierarchy.

An end user can visually track what effect a modification on one node will have on the rest of the hierarchy. For example, the user may want to change *Shipping Distance* from 1000 miles to 2000 miles and observe what affect this change will have, in sequence (from bottom to top), on *Distance Range,* the *Haul Type,* and ultimately the *Transportation Mode* recommendation itself. The upward pointing links indicate that the user may change a value on a lower-level node and watch how the modification propagates upwards. In other words, forward chaining takes place, and the end-user can visually observe how values on a lower-level can subsequently change the values of higher-levels of the tree structure, level-by-level.

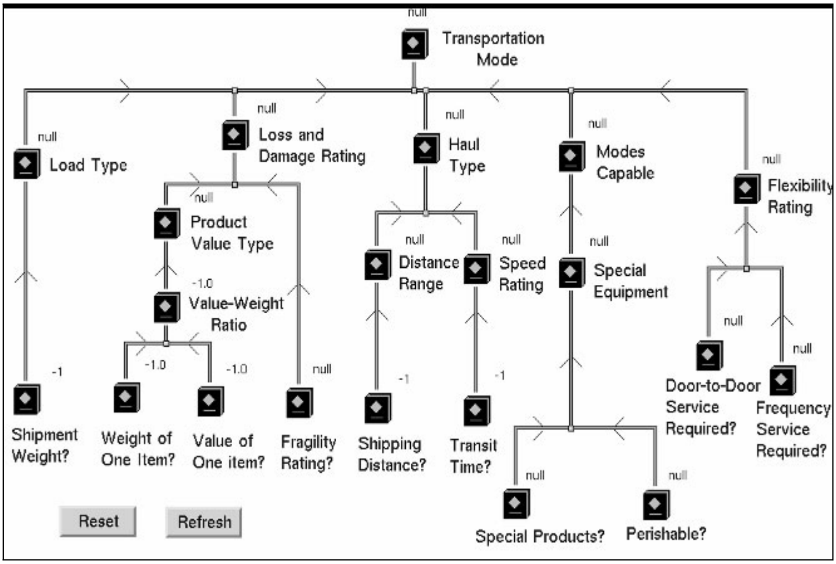


Figure 7.1. Hierarchic model of Expert-Strategy

The hierarchic models are intended to provide explanations as to why a given conclusion was reached. For instance, an end user may disagree with a conclusion and may want to understand what input was the cause of the conclusion. Through an exploration of the components of the hierarchy, the end user can determine what input variable(s) was responsible for the system recommendation.

7.5.2 Restrictive vs. Non-restrictive Systems (Selection of an Advisory Strategy)

LogNet (Nakatsu 2005) is an intelligent system that offers advice on how to design business logistics networks. It attempts to design the most cost-effective network satisfying a certain customer service level. It addresses the facility location problem: How many warehouses are needed in a logistics network? Where should these warehouses be located and to which customer markets should they serve? (Ballou 1992).

At the heart of LogNet is the logistics network model (see Figure 7.2). Three types of nodes are considered: factories are where the product is manufactured; warehouses receive the finished product from the factories for storage and possibly for further processing; customer markets place orders and receive the desired products from the assigned warehouse. Product moves through the logistics network via transportation links: inbound links move product from factory to warehouse and outbound links move product from warehouse to customer market.

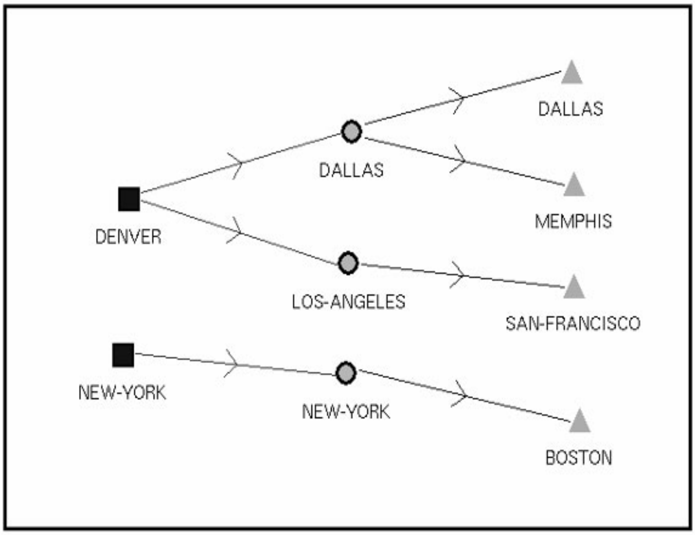


Figure 7.2. A network configuration model of a logistics environment

Different network configurations can be created and tested by the end user, and such performance measures as costs (inventory, transportation, and warehousing) and customer service level (customer service is defined as the average distance between the customer market and the assigned warehouse) can be calculated by the system at the end-user’s request.

In order to offer intelligent advice, LogNet employs problem-solving procedures (hereafter referred to as “operators”) that actual business experts of business logistics might use to design business logistics networks. The seven operators that LogNet uses are as follows:

1. **Completeness:** Checks whether the network is complete. A network is complete if all customer demand has been assigned to valid warehouses, and all warehouse demand has been assigned to valid factories.
2. **Outbound Links:** Checks whether the inbound links are the closest distance possible.
3. **Inbound Links:** Checks whether the outbound links are the closest distance possible.
4. **Consolidation 1:** Suggests the consolidation opportunity that will result in the greatest cost savings.
5. **Consolidation 2:** Suggests the consolidation opportunity that will result in a cost saving, but done so with least damage to customer-service levels.
6. **Decentralization 1:** Suggests the decentralization opportunity that will result in the greatest improvement to customer-service levels.
7. **Decentralization 2:** Suggests the decentralization opportunity that will result in an improvement to customer-service levels, but achieved at the lowest possible cost.

LogNet implements its capabilities by utilizing model-based reasoning techniques. For example, LogNet employs model-based reasoning to analyze the structure of the current logistics network, and evaluates all possible consolidation opportunities between two warehouses. The warehouse location problem can also be tackled using optimization techniques well-known in Operations Research. For example, linear programming techniques can find the minimum network costs satisfying a variety of constraints (*e. g.* customer-service level must be at least a certain level).

However, such methods, which utilize precise analytical methods to evaluate alternatives, frequently compromise too much in terms of flexibility and realism of the problem-solving situation (Simon 1996). On the other hand, visual interactive modeling (Angehrn and Luthi 1990) is better suited to more open-ended design tasks, because they provide guidelines to reason about a network structure (in an intuitive way), rather than hard-and-fast analytical methods. This approach offers an alternative to optimization methods if one wishes to have more flexibility in terms of problem-formulation and the incremental testing and design of different network design. Their major disadvantage, however, is that they do not guarantee optimal solutions.

Two versions of LogNet were created to manipulate system restrictiveness: the first version, **Free Form LogNet**, is a non-restrictive version; the second, **Restrictive LogNet**, is a restrictive version. By system restrictiveness we are referring to the extent to which a system limits the manner in which the system is used. Whereas Free Form LogNet allows a user to request any of the seven operators *in any order and at any point* during a consultation, Restrictive LogNet decides the manner in which the seven operators are requested. In fact, users of Restrictive LogNet are unaware of the seven operators. Moreover, the seven operators are user-invoked and discretionary under Free Form LogNet, whereas they are automatically provided in Restrictive LogNet.

In an experimental study conducted (Nakatsu and Benbasat, forthcoming 2005), we found that Restrictive LogNet was beneficial for structured tasks that arrived at the correct solution—users were more liable to engage in incorrect problem-solving behaviors under Free Form LogNet because its open-ended interaction offered very little guidance on how to use the seven operators. On the other hand, Free Form LogNet was more helpful for unstructured tasks in which some type of failure occurs—that is, the system arrives at an incorrect or suboptimal solution, in which case the user must "see" beyond the system's recommendations. Free Form LogNet users were more proficient at these tasks because they had more direct control over the seven operators, and a more thorough understanding of the way LogNet worked via the seven operators. The non-restrictive nature of the advisory strategy forced these users to explore the system and understand it more fully.

7.5.3 Deep Explanations (Content-based Enhancement)

Expert-Strategy also provides deep explanations (Nakatsu and Benbasat 2003) that explain the underlying domain principles associated with a given recommendation. In the transportation mode problem discussed above in Section 7.5.2, the deep explanations reflect the underlying domain principles in transportation and logistics that are associated with the rules. The deep explanations are dynamically generated whenever the system reaches a transportation mode recommendation.

For example, when the following rule is executed, Expert-Strategy concludes that TRUCKING is the transportation mode:

If load-type = (very-large-shipment or large-shipment) AND haul-type = medium-fast
THEN transportation-mode = TRUCKING

Displaying the above rule would not be very informative for an end-user trying to understand the reason for a system recommendation. By providing a deep explanation, the user obtains a better understanding as to why the system selected TRUCKING. The deep explanation is given in Figure 7.3.

The first fragment of the deep explanation is based on the value given by **load type.** Based on load type alone, Expert-Strategy narrows the candidate choices to RAIL and TRUCKING. A domain principle is then provided explaining why the system has narrowed the decision to two options. The second fragment is based on the value given by **haul type.** Expert-Strategy further narrows the decision by selecting TRUCKING, and gives its associated explanation.

DEEP EXPLANATION OF THE SYSTEM RECOMMENDATION

Load type:
For large to very large shipment sizes, RAIL and TRUCKING are the preferred transport modes. RAIL tends to move large and very large shipment sizes of at least a full carload (greater than 15,000 lbs.), due to the heavy load capability of railroads, as well as the cost structure associated with railroads (*i. e.*, high fixed costs and low variable costs). TRUCKING is the most versatile of the shipment modes handling small, average, large, and very large loads. AIR service, on the other hand, is generally restricted to average and small shipments since it is more constrained by the physical dimensions of a cargo plane.

Haul type:
In choosing between RAIL and TRUCKING, if haul type = medium-fast, then the transport mode of choice is TRUCKING. This is because, between the two choices, TRUCKING can move cargo faster than RAIL over medium-distance hauls

Figure 7.3. Sample deep explanation

7.6 The Three Faces of Explanatory Power: A Research Framework

The emphasis of this chapter has been on the three types of enhancements to explanatory power: content, characteristics of the user interface, and advisory strategies. Figure 7.4 depicts the framework, which includes both the enhancements and outcomes of explanatory power.

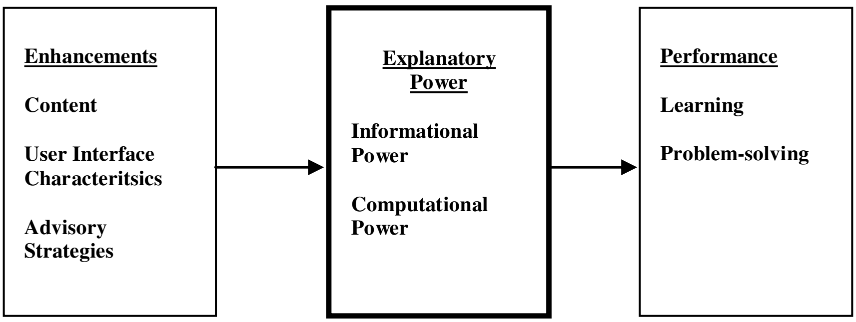


Figure 7.4. A Framework of the explanatory power of a user interface

Two different aspects of explanatory power are included in Figure 7.4, informational power and computational power of a user interface. These concepts can be used to evaluate two different user interfaces. Larkin and Simon (1987), in their discussion on comparing diagrammatic representations, define two representations as **informationally equivalent** "*if all the information in the one is also inferable from the other, and vice versa*" (p. 70). Moreover, two representations are **computationally equivalent** "*if they are informationally* *equivalent and, in addition, any inference that can be drawn easily and quickly from the information given explicitly in the one can also be drawn easily and quickly from the information given explicitly in the other, and vice versa*" (p. 70).

From these definitions, we derive the notion of informational power, which is strictly related to content alone, while computational power evaluates efficiency and usability as well. Interfaces having greater informational power contain superior content, and interfaces having greater computational power have superior content that can be used and accessed more efficiently and effectively. Two user interfaces are said to be equivalent, in terms of explanatory power, if they are computationally equivalent (which, by Larkin and Simon's definition, means they are informationally equivalent as well).

Let us consider how each of the three determinants of explanatory power can affect informational power and computational power. Content-based enhancements, strictly speaking, can only increase informational power of the user interface. Interface-based enhancements and selection of an appropriate advisory strategy can also increase the computational power of a user interface. In summary, explanatory power of a user interface, then, is a more comprehensive construct than explanations content alone, because it also considers gains to be achieved in computational power as well.

It is worthwhile to point out that the three determinants of explanatory power—content, user interface, and advisory strategy—can often interact with one another in interesting ways to increase system performance. While it is useful to separate out the three determinants individually, in actual practice it is often difficult to speak of one determinant, in isolation, as creating more explanatory power. In fact, two determinants frequently work in tandem to create more explanatory power, so that it is often difficult to determine where one determinant ends and the other begins. For example, the support for graphical representations in the user interface may allow for the development of hierarchic strategic models that may be more powerful than the specification of strategic knowledge through a text-based interface. In this particular instance, both the interface, and the actual content of the system are enhanced. Similarly, the ability to inspect the components of a system may result in an interface that allows for more powerful deep justifications, which would not be possible in a more static user interface. Still another example would provide for appropriate feedback advice in a timely manner, in which content is dynamically generated and case specific (interaction of advisory strategy and content).

Enhancing the explanatory power of intelligent systems can result in systems that are easier to use, and result in improvements in decision-making and problem-solving performance. Two experimental studies have been conducted that show promising results. In one, we developed hierarchic models of an Expert-Strategy that were intended to help a user to better understand the way that an expert system reasons (Nakatsu and Benbasat 2003). Users overwhelmingly preferred this type of interface, which gave them the ability to visualize how an Expert System reasons. In a second study (Nakatsu and Benbasat, forthcoming 2006), we varied the advisory strategy of a decision support system that aided users in designing business logistics networks. As discussed above, two special modes of operation were tested: a restrictive system that provided structured advice versus a non-restrictive system that was more open-ended and allowed users to request the problem-solving procedures in any order they wished. We found that the restrictive system was more effective for structured tasks (not surprising), but that the non-restrictive system helped users deal more effectively with novel problem-solving situations in which some type of system failure occurs (*i. e.* the system generated suboptimal advice). Selection of an appropriate advisory strategy proved crucial.

Enhanced explanatory power can also result in more learning and better understanding. Schank's (1986) view of explanation as a process of integrating and assimilating new information suggests that explanations serve the role of developing more powerful mental models of the world. User interfaces having explanatory power can foster an assimilative learning process in which the user is actively engaged in developing powerful mental models of a system. The deep explanations described in Section 7.5.3, for example, were crafted to foster learning and understanding of an expert system (as opposed to having just a shallow understanding of the system). In the experimental study of Expert-Strategy, we devised a problem-solving task in which participants were required to have a deeper understanding of an expert system rule base. Subjects were instructed to identify portions of a rule base using faulty logic and reaching incorrect conclusions.To answer this question correctly, a subject would presumably need to have an appropriate mental model of how the rule base is reaching system conclusions. We found strong support that the provision of deep explanations led to better understanding and learning: subjects provided with the deep explanation support were much more likely to answer this question correctly.

7.7 Intelligent Systems Today

There appears to be a widely held perception that much of the well-publicized successes of intelligent systems, in particular expert systems, have been more hype than actual gains in workplace performance, and that the great potential of intelligent systems has just not panned out as anticipated. In line with this perception, Gill (1995) observes that several AI vendors have failed, and major companies have become disillusioned with the technology, many reducing or even eliminating their commitment to the technology altogether.

Despite the widespread recognition of the technology's limitations and weaknesses, there have been many notable successes of expert systems use in commercial applications. The rate of application to the larger commercial world was dramatic in the 1980s: during this time period, Durkin (1996) estimates that over two-thirds of the Fortune 1000 companies applied expert-systems technology to daily business applications. Eom (1996) surveyed publications on operational expert systems from 1980-1993, and found that many expert systems have had a profound impact on organizations, in some cases shrinking the time for tasks from days to hours, minutes or seconds. In addition, he found that there were many nonquantifiable benefits such as improved customer satisfaction, improved quality of products and service, and more consistent decision-making.

A recent survey of expert system shells turned out over 60 different products (Commercial Expert System Shells 2005). These products ranged from very inexpensive shells running on PCs costing less than $100, to very powerful, industrial-strength shells running on powerful workstations and mainframes costing several thousands of dollars. There are several noteworthy trends regarding these products, which can serve to illustrate the direction the industry has been moving in:

- The shells incorporate several newer AI techniques, other than rule-based reasoning. For example, case-based reasoning (*e. g.* CBR Express, CPR, The Easy Reasoner), fuzzy logic (*e. g.* EXSYS), neural networks, genetic algorithms, model-based reasoning (*e. g.* Gensym's G2), frame-based reasoning (*e. g.* FLEX) and other techniques.
- The newer and more powerful shells include support for the development of powerful graphical user interfaces. RTWorks enables graphical objects that can be tied to variables that dynamically control attributes such as color, scale, rotation, motion, animation, *et al.* Gensym's G2 enables the development of graphical object models to capture structural information about the application domain (*e. g.* a hydraulic system, a business process, a nuclear power plant).

Expert-systems technology, as evidenced by the shells out on the market today, is developed, more and more frequently, as embedded technology within larger applications, rather than as standalone expert systems. This observation is in line with many survey articles on the state of expert systems technology (Durkin 1996, Hayes-Roth and Jacobstein 1994, p. 27, Liebowitz 1997). In part, this means that these shells must include capabilities other than rule-based reasoning. Indeed, as the above survey shows, many of these shells come equipped with powerful graphical modeling environments, provide support for integration into other environments, and utilize a variety of AI techniques, other than just rule-based reasoning.

All in all, we think the investigation of explanatory power is a fruitful area of research, one that is more in line with the most important issues facing designers and researchers of intelligent systems today. For the designers of intelligent systems, our framework provides design guidelines that can be used to enhance explanatory power, and create more powerful graphical user interfaces. This is an especially important issue today given that systems are more complex and heterogeneous. For researchers, our framework builds on and extends the work that has been accomplished on explanations research, which has a long tradition in artificial intelligence and human-computer interaction.

References

Aikens J (1983) Prototypical knowledge for expert systems. Artificial Intelligence, 20: 163-210.

Angehrn AA, Luthi H (1990) Intelligent decision support systems: a visual interactive approach. Interfaces, 20(6): 17-28.

Ballou RH (1992) Business Logistics Management, Third Edition, Englewood Cliffs, NJ: Prentice Hall.

Brown JS, Burton RR, de Kleer J (1982) Pedagogical, natural language, and knowledge engineering techniques in SOPHIE I, II, and III, in Sleeman D, Brown JS (Eds.), Intelligent Tutoring Systems, New York: Academic Press, pp 79-98.

Buchanan BG, Shortliffe EH (1985) Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project, Reading, MA: Addison-Wesley.

Carroll JM, Carrithers C (1984) Training wheels in a user interface. Communications of the ACM, 27 (8): 800-806.

Carroll JM, McKendree J(1987) Interface Design Issues for Advice-Giving Expert Systems. Communications of the ACM, 30(1):14-31.

Chandrasekeran B, Tanner MC, Josephson JR (1988) Explanation: The Role of Control Strategies and Deep Models, in Hendler JA (Ed.), Expert Systems: The User Interface,. Pp 219-247.

Clancey WJ (1983) The epistemology of a rule-based expert system—a framework for explanation. Artificial Intelligence, 20: 215-251.

Clancey WJ, Letsinger R (1981) Reconfiguring a rule-based expert system for application to teaching, in Proceedings of the 7th International Joint Conference on Artificial Intelligence, 20: 215-251.

Commercial Expert System Shells (2005), http://www-2.cs.cmu.edu/Groups/AI/html/faqs/ai/expert/part1/faq-doc-7.html

Davis R, Hamscher W (1988) Model-based reasoning: troubleshooting, in Shrobe H (Ed.), Exploring Artificial Intelligence, San Mateo: Morgan Kaufmann Publishers, pp 297-346.

Dhaliwal JS, Benbasat I (1996) The use and effects of knowledge-based systems explanations: theoretical foundations and a framework for empirical evaluation. Information Systems Research, 7 (3): 342-362.

Durkin J (1996) Expert systems: a view of the field. IEEE Expert, 56-63.

Ellis C (1989) Explanation in intelligent systems, in Ellis C (Ed.), Expert Knowledge and Explanation: The Knowledge-Language Interface, New York: Halstead Press, pp 103-126.

Eom, SB (1996) A survey of operational expert systems in business (1980-1993). Interfaces, 26(5): 50-70.

Fischer B, Lemke A, Schwab T (1985) Knowledge-based Help Systems, in Proceedings of CHI'85 Human Factors in Computing Systems, San Francisco, CA, pp 161-167.

Franklin S (1997) Artificial Minds, Cambridge, MA: MIT Press.

Gensym Corporation (1997) G2 Reference Manual, Cambridge, MA: Gensym Corporation.

Gill TG (1995) Early expert systems: where are they now? MIS Quarterly, 19(1): 51-81.

Gregor S, Benbasat I (1999) Explanations from intelligent systems: theoretical foundations and implications for practice. MIS Quarterly, 23(4): 497-530.

Hasling DW, Clancey WJ, Rennels G (1984) Strategic explanations for a diagnostic consultation system. International Journal of Man-Machine Studies, 20: 3-19.

Hayes-Roth F, Jacobstein N (1994) The state of knowledge-based systems. Communications of the ACM, 37(3): 27-39.

Kunz JC (1987) Model Based Reasoning in CIM, in Intelligent Manufacturing. Proceedings from the First International Conference on Expert Systems and the Leading Edge in Production Planning and Control, pp 93-112.

Larkin JH, Simon HA (1987) Why a Diagram is (Sometimes) Worth Ten Thousand Words. Cognitive Science, 8: 255-273.

Liebowitz J (1997) Worldwide perspectives and trends in expert systems. AI Magazine, Summer 1997:115-119.

Jagodzinski AP (1983) A theoretical basis for the representation of on-line computer systems to naive Users. International Journal of Man-Machine Studies, 18(3): 15-252.

Mockler RJ (1989) Knowledge-Based Systems for Management Decisions, Englewood Cliffs: NJ: Prentice-Hall.
Moffit KE (1994) An analysis of the pedagogical effects of expert system use in the classroom. Decision Sciences, 25(3): 445-460.
Moffit KE (1989) An Empirical Test of Expert System Explanation Facility Effects on Incidental Learning and Decision-making, Unpublished Doctoral Dissertation, Arizona State University.
Nakatsu RT (2001) Enhancing the Explanatory Power of Intelligent, Model-Based Interfaces, Unpublished Doctoral Dissertation, The University of British Columbia.
Nakatsu RT (forthcoming, 2005) Designing business logistic networks using model-based reasoning and heuristic-based searching. Forthcoming in Expert Systems with Applications, 2005.
Nakatsu RT, Benbasat I (2003) Improving the explanatory power of knowledge-based systems: an investigation of content and interface-based enhancements. IEEE Transactions on Systems, Man, and Cybernetics, Part A, 33 (3): 344-357.
Nakatsu RT, Benbasat I (forthcoming, 2006) Designing intelligent systems to handle system failures: enhancing explanatory power with less restrictive user interfaces and deep explanations. Under review in International Journal of Human-Computer Interaction.
Patil RS (1988) Artificial intelligence techniques for diagnostic reasoning in medicine, in Shrobe H (Ed.), Exploring Artificial Intelligence, San Mateo: Morgan Kaufmann Publishers, pp 347-380.
Schank RC (1986) Explanation: A First Pass, in Kolodner JL, Riesbeck CK (Eds.), Experience, Memory, and Reasoning, Hillsdale, NJ: LEA, pp 139-165.
Simon HA (1996) The Sciences of the Artificial, Third Edition, The MIT Press, Cambridge, MA.
Southwick RW (1988) Topic Explanation in Expert Systems, in Kelly B, Rector A (Eds.) Research and Development in Expert Systems V, Proceedings of the Expert Systems '88, pp 47-57.
Southwick RW (1991) Explaining reasoning: an overview of explanation in knowledge-based systems. Knowledge Engineering Review, 6(1): 1-19.
Stelzner M, Williams MD (1988) The evolution of interface requirements for expert systems, in Hendler JA (Ed.), Expert Systems: The User Interface, pp 219-247.
Swartout WR (1983) XPLAIN: A system for creating and explaining expert consulting programs. Artificial Intelligence, 21: 285-325.
Todd FJ, Hammond KR (1965) Differential effects in two multiple-cue probability learning tasks. Behavioral Science, 10: 429–435.
Turban E (1995) Decision Support and Expert Systems, Fourth Edition, Englewood Cliffs, NJ: Prentice Hall.
Wallis JW, Shortliffe EH (1985) Customizing explanations using causal knowledge, in Buchanan BG, Shortliffe EH (Eds.), Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project, Reading, MA: Addison-Wesley, pp 371-388.

Part II

Applications of Intelligent Decision-making Support Systems

8

A New Paradigm for Developing Intelligent Decision-making Support Systems (i-DMSS): A Case Study on the Development of Comparison-Shopping Agents

Yun Wan

Department of Computer Science, University of Houston, Victoria, 4902 Mulberry Circle, Missouri City, TX 77459, USA, e-mail: wany@uhv.edu

There are well-documented approaches and methodologies for information system development (ISD). These approaches and methodologies can all be related to software development life cycle (SDLC), a paradigm focusing on ISD in organization and other closed-domain systems. Recent developments in agent-oriented software engineering, like Gaia, extend this paradigm to cover intelligent agents and multiagent systems. However, evidence from Web-based i-DMSS like comparison-shopping agents shows distinctive patterns of development. Since there are few studies that cover the ISD paradigm for agents in an open-domain system, through three mini case studies in this chapter, we provide relatively comprehensive cases for future theory formulation.

8.1 Introduction

Comparison-shopping agents are those emerging Web-based intelligence information systems that can collect product and service information – especially price-related information – from multiple online vendors, aggregate them, and then process them into value-added information for online shoppers to assist their decision making. In the popular press, comparison-shopping agents are also called Shopbots, aggregators, or, simply, Bots.

Comparison-shopping agents are intelligent decision-making support systems (i-DMSS) because: Intelligence is the ability to learn or understand or to deal with new situations. When we apply this to software agents, it refers to the agent's ability to accept different users' statement of goals and carry out the task delegated to it (Bradshaw 1997). Since comparison-shopping agents can accept the search request for products and services from online shoppers and then retrieve relevant information, they show a certain level of intelligent behavior. So comparison-shopping agents are intelligent software agents.

Decision-making support systems (DMSS) or Decision Support Systems (DSS) were originally defined as interactive computer based systems that can help decision makers utilize data and models to solve unstructured problems (Sprague 1980). Comparison-shopping agents allow the decision maker (online shopper) to deal with a specific set of related problems (shopping decisions), so they are *ad hoc* decision support systems and can be classified as "Specific DSS" according to Sprague (1980).

The development paradigm of comparison-shopping agents is the focus of this research. To give a comparative analysis on the new paradigm adopted for these agents, we first give a review of the system development life cycle (SDLC).

8.1.1 An Overview of SDLC

Almost all existing ISD approaches and methodologies can be grouped under one paradigm characterized by the system development life cycle (SDLC).

SDLC is a canopy name for a group of similar and historically connected methodologies used in information system development. The basic structure of SDLC starts with an initial feasibility study, going through stages like system analysis and design, implementation, and finally entering into the stage of maintenance. Depending on the evolution of software engineering techniques being adopted in system design, ISDs can be categorized into three classes, and each class dominated a certain stage of the history.

The earliest SDLC methodologies were proposed in the 1970s and were characterized by process-centered software-engineering approaches. They include the waterfall model, which was the original SDLC method (Royce 1970); the joint application development (JAD) model, which was developed by Chuck Morris of IBM Raleigh and Tony Crawford of IBM Toronto in 1977; and the rapid application development (RAD) model (Martin 1991). These models are compatible with structured programming. In the 1980s, new methodologies like the fountain model (Henderson-Sellers and Edwards 1993), the spiral model (Boehm 1988), and the stakeholder win-win Model (Boehm *et al.* 1995) were developed to adapt to object-oriented software engineering techniques. In the past 5 to 10 years, Gaia (Wooldridge *et al.* 2000) and other nascent agent-based ISDs have been proposed to accommodate increasingly complex organizational environments. These ISDs were based on agent-oriented software engineering techniques. In the next section, we briefly described the features of these methodologies in the SDLC paradigm.

8.1.1.1 From Process-centric to Object-oriented

Early ISD approaches like the waterfall model were process-centric, which was a natural fit for structured programming techniques. For example, the waterfall model emphasized the steady developmental flow from requirement analysis to system integration and maintenance. The development cycle of the Waterfall model was often extended due to the time each development stage required.

Because all subsequent stages are dependent on effective system analysis and design, the Waterfall model places very high stakes on requirement analysis, which made the initial stage of development very influential on the success of project. As a result, considerable effort had to be betted upfront (Figure 8.1).

The deficiency of the Waterfall model became obvious when the system was being developed in a complex and rapidly changing environment or when the users or stakeholders of the system had no specific requirements at the beginning of the project.

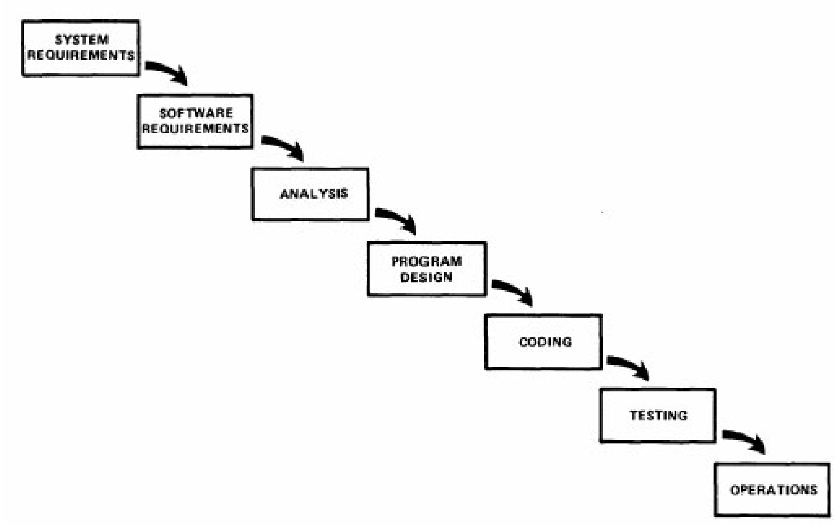


Figure 8.1. The waterfall model (Royce 1970)

The improvements to overcome the disadvantages of the Waterfall approach thus mainly go into two directions: reducing the time needed for each development cycle and balancing the risk of betting upfront. JAD and RAD were two approaches in these directions.

The JAD approach aimed at balancing the risk of betting upfront by emphasizing heavy user involvement in the whole development process. In early JAD models, users were required to stay at the same physical location with developers to facilitate collection and analysis of user requirements. By keeping the user involved, many design problems could be detected and corrected early to reduce the risk of project failure. Some researchers regarded JAD as part of the practice of RAD, which was to draw users into the development process (Robinson 1996).

The RAD approach emphasized iterative, evolutionary prototyping and aimed at reducing the time needed for the development cycle by providing a standardized development framework and tools for the project. In the RAD approach, focus-group sessions are being used to give both developers and users a channel to communicate and clarify project requirements and design improvements. These sessions are coordinated by experienced facilitators to guarantee the quality of communication.

The RAD approach allowed a project to be quickly developed though it had to sacrifice some level of flexibility and efficiency (Martin 1991).

When entering the 1990s, the business environments were increasingly complex. Technology development and turbulent organizational competition landscaped demand to a new level of requirements for efficiency and effectiveness of SDLC. To accommodate the requirements, the RAD approach was further refined, and the paradigm in general moved to contingency design.

The contingency idea was not new. When the complexity of the information system could no longer be accommodated by process-centric structural development, information science researchers began to explore the contingency ways of system development, which can be traced back to the 1970s (Berrisford and Wetherbe 1979, Sprague 1980, Alavi and Henderson 1981).

Prototyping or heuristic methodologies as one contingency solution were refined in this background, a "quiet revolution" in the information industry according to Naumann and Jenkins (1982). Not surprisingly, the most frequently and successfully applied domain for prototyping or heuristic development approaches was on ad hoc decision support system. The mini-case study in Berrisford and Wetherbe (1979) vividly described how a DBMS system used by an oil-exploration firm was developed first as a prototype and then by adding component by component according to the requirements of geophysical personnel. In retrospect, it was obvious that because this was a highly specialized system and system analysts lacked the necessary domain knowledge, they had to or were forced to use this approach.

Parallel to prototyping, the iterative concept was even more sophisticated in contingency design. The fountain model (see Figure 8.2) and Spiral model are such approaches.

The Fountain model use the same analogy of the waterfall model, here the water flow is a fountain and can rise up to the middle and fall back, either to the "pool" below or re-enter at an intermediate level. In other words, the iteration can happen at any stage and go back to any stage of development (Henderson-Sellers and Edwards 1993).

The spiral model was more famous due to its wide publicity. It twisted the fountain water flow into a spiral water flow and emphasized the integration of design and prototyping in each development stage. By iterating design and prototyping in each stage, the system can update its deficiency in early design and accommodate new user requirements in a timely way (Boehm 1988) (Figure 8.3).

The more up-to-date SDLC approach in this direction was an upgraded version of the spiral model called the stakeholder win-win spiral model. This approach extended the spiral model by adding Theory W activities to the front of each cycle. Theory W argued that making the system's key stakeholders winners was a necessary and sufficient condition for project success (Boehm and Ross 1988 1989). So the stakeholder win-win spiral model added front-end activities (Figure 8.4 shaded part) such as identify system key stakeholder, win condition of stakeholders and reconciliation of stakeholders' win conditions. These activities demonstrate where objectives, constraints, and alternatives come from for the project. This lets users clearly identify the rationale involved in negotiating win conditions for the project.

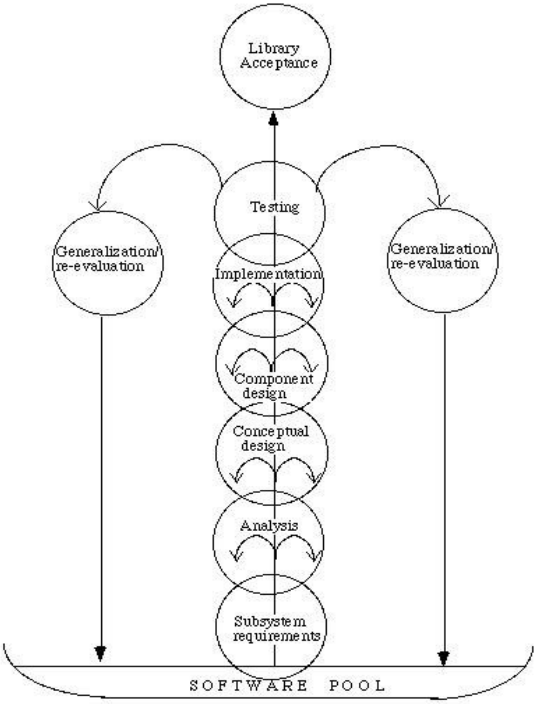


Figure 8.2. The fountain model (Henderson-Sellers and Edwards 1993)

A new extension of SDLC, agent-oriented software engineering (AOSE) has become popular since 2000. In AOSE, agents instead of objects are used as the basic unit of design of the system, which provided us two advantages:

First, an agent could support an advanced level of interaction with other agents like those described in "speech act" theory compared with *ad hoc* messages used between objects. Secondly, agents have autonomous behavior, which may not necessarily be controlled from the outside, while objects have to be controlled from the outside.

These two features made AOSE a more appropriate methodology in coping with highly complex environments compared with object-oriented approaches. In addition, all these formally published AOSE methods had an emphasis on some particular stage of SDLC. For example, Tropos emphasizes the early requirements analysis stage (Mylopoulos *et al.* 2001); Gaia emphasizes the system-analysis and design stages (Wooldridge *et al.* 2000); MaSE emphasizes the implementation stage with automatic code generation (Wood and DeLoach 2000).

Though different in focus, most development in AOSE for the time being is on the theoretical stage and many of them were still focusing on a closed domain like an organization instead of an open domain like Internet. We used Gaia as an example.

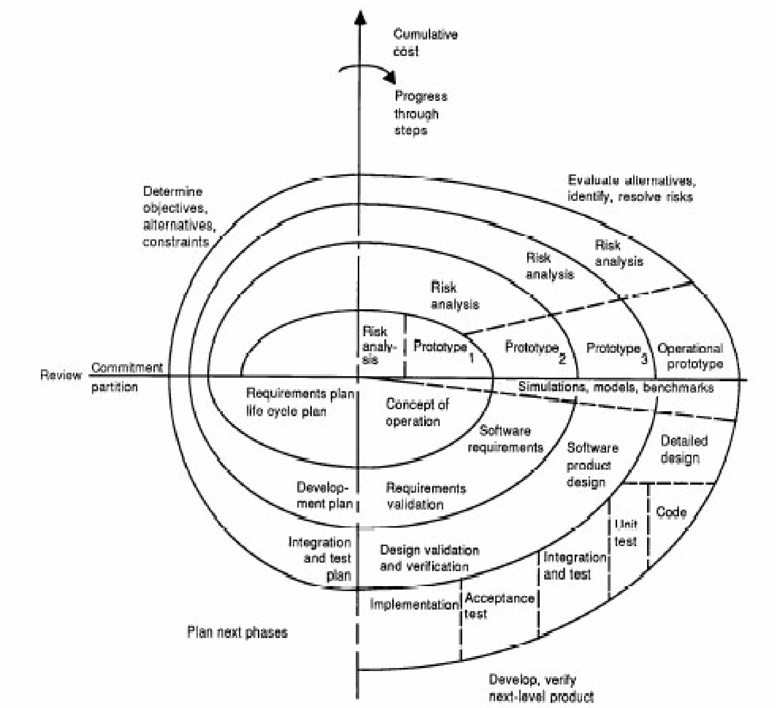


Figure 8.3. The spiral model (Boehm 1988)

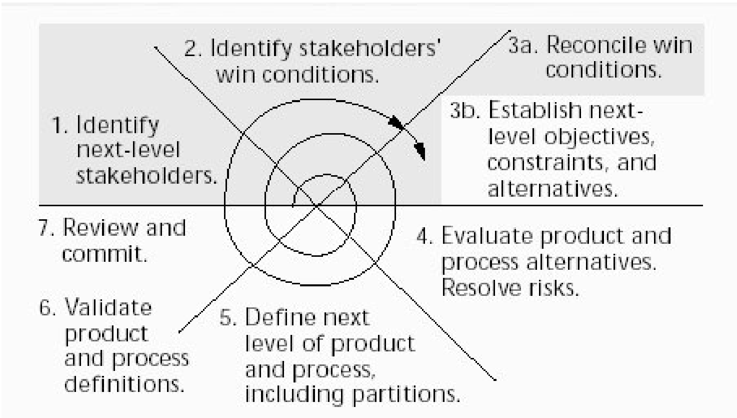


Figure 8.4. Stakeholder win-win spiral model (Boehm *et al.* 1995)

Gaia was one of the earliest agent-oriented ISDs with an emphasis on system analysis and design. In Gaia, the whole system analysis and design process was re-organized around different agent-based models, which included role models and interactions models for agents in the analysis stage and agent models, service models, as well as acquaintance models in the design stage (Figure 8.5).

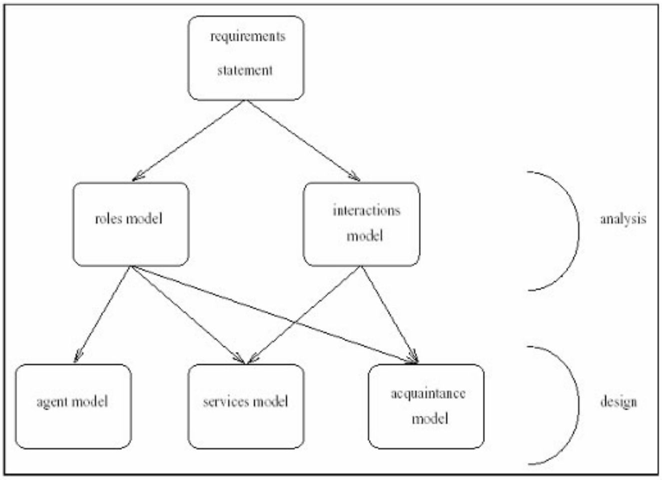


Figure 8.5. Relationships between the methodology's models (Wooldridge *et al.* 2000)

Compared with traditional ISDs, where the analysis and design process was developing abstract models and transforming them into a sufficiently low level of abstraction for implementation, Gaia's analysis and design process was focusing on how to transform abstract models developed in the analysis stage into a sufficiently low level of abstraction to implement them into agents.

The basic features of Gaia indicated its origin from SDLC. Though in many aspects, it is a step advanced from traditional ISDs in coping with complex environments, it still assumed that there existed a life cycle of the agent system and two distinctive stages for system analysis and design.

More importantly, when coping with complex environment, agent-based approaches are still dependent on more sophisticated technologies to internalize the conflict between the rational core of the system and outside changing environment instead of utilizing the opportunities provided by the environment. For example, Gaia tried to through the use an agent, a more sophisticated technology unit, instead of an object to increase the flexibility and autonomy of the system to deal with the changing environment, the primary source of complexity. However, from what we observed in the ISD for comparison-shopping agents, there is a different way of handling such conflicts, as will be illustrated.

The above discussion indicates the basic evolution trend for SDLC paradigm, which includes more effective requirement analysis, sophistication of software engineering technologies (from object oriented to agent oriented), and more elaborated iterations during different stages of the cycle (from JAD to spiral model).

These evolution features may match the complexity requirements for close domains like organizational information systems because they had designated design goals, prespecified development schedules, and clearly identified development stages. However, it may be problematic for i-DMSS in open domains like the Internet. Next, we first describe the ISD features of comparison-shopping agents and then show its difference from traditional SDLC paradigm.

8.1.1.2 The New Features of the ISD of Comparison-Shopping Agents

Comparison-shopping agents are Web-based *ad hoc* i-DMSS systems being used by the general (online) public to obtain online shopping information. Because they were information intermediaries between online vendors and online shoppers, they have much less self-control on the timing of their development and maintenance cycle. Instead, they have to be highly adaptive to the environment.

A comparison-shopping agent has to be as stable as possible because the service they provide needs to be accessed at any time. Any glitch in service would cause huge losses for service providers. So it does not allow for strategies like prototyping. Comparison-shopping agents also have no specific development schedule. They have to constantly adapt to the changes in online shopping environment, so they have to be developed on demand, though the design goal is usually unknown beforehand. A comparison-shopping agent needs to deal with competition from peer agents by providing comparable services. This leads to imitation, merging, acquisition as well as the implementation of other strategies into development.

In general, the environment for a comparison-shopping agent was far more complex than those for traditional i-DMSS. If we compare the system development for comparison-shopping agent with i-DMSS in a closed domain, we can find that:

First, there was no specific user involved in requirement analysis for the design of comparison-shopping agent. Since comparison-shopping agents are designed for the general public. It was impossible to accommodate a specific user's preference. As a result, the designer has to formulate the initial requirements based on intuition or imitate other established agents. Actually imitation has replaced requirement analysis for the development of most comparison-shopping agents.

Secondly, there is no clear stage boundary between system design and implementation. Because comparison-shopping agents were competing for the same users (online shopper), they have to incorporate new features constantly. As a result, the system design and implementation were not separated at all.

Thirdly, there is no clear boundary between system implementation and maintenance. Since comparison-shopping agents are placed in an open environment for use, they must be ready for any new development based on changing demands from online shoppers or pressures from their innovative peers. So there was no completion of the system. There were only irregular intervals between system maintenance and further development. In other words, there was no life cycle for comparison-shopping agents.

Table 8.1. Comparison between SDLC paradigm and new paradigm

	SDLC Paradigm	**New Paradigm**
Requirement Analysis	Defined by Users	Defined by Imitation
System Design	Encourage close developer-user interaction	No interaction needed
System Implementation	Controlled by system user or stakeholders	Contingent on environment
System Maintenance	Clearly separated from implementation	Not separated from implementation

The complex environment of comparison-shopping agents constitutes considerable challenges for the design and development of these agents; as a result, the new paradigm for Web-based agent development needs to be investigated. Here we identify the feature of this new paradigm adopted by comparison-shopping agent developers. We demonstrate this new paradigm through three mini case studies, and based on that, we try to illustrate in detail the traits for this new paradigm.

8.2 Three Mini Case Studies on the Development of Comparison-shopping Agents

Here we present three mini case studies about comparison-shopping agents. Through these three cases, we want to demonstrate the impact of environments on the development approach/methodologies/strategies of i-DMSS in the open domain. We also want to show that for information intermediaries like comparison-shopping agents, system development strategies were usually indistinguishable from and interweaved with business strategies.

The first case study discussed how one of the earliest comparison-shopping agents, BargainFinder.com, was successfully developed in a relatively simple environment via pure technical sophistication.

The second case discussed is about how another of the earliest comparison-shopping agents, Pricewatch.com, achieved the same performance by using both technical strategies and business strategies and then become prosperous today.

The third case study compares the development patterns of the luck luster performance of Pricescan.com and the rather competitive performance of Shopping.com. We analyze how this new generation of comparison-shopping agents, Shopping.com, overcame the disadvantages of environment factors and grew into top players in the B2C electronic market.

8.2.1 Case Study 1: The Instant Fame of BargainFinder

In 1995, Jeff Leane, Bruce Krulwich, and several other researchers in Andersen Consulting's research lab and smart store center built BargainFinder for automatic price gathering and comparison shopping for music CDs. BargainFinder was the first generation of the comparison-shopping agents that appeared in the US. It was developed as an online experiment to measure the reactions of consumers and merchants to price comparison provided by Web-based intelligent decision support systems.

BargainFinder can take the artist or album name for a CD and then search nine Internet music stores. It then returned a list of the prices found. The online shopper can then select the preferred online store and be taken directly into that album page in the store website.

BargainFinder was a typical information integration system. It restructures the user query into nine specific queries acceptable by those nine online music stores and then extractes the response from those stores, integrates them and presents the integrated comparison-shopping list to the user (Figure 8.6).

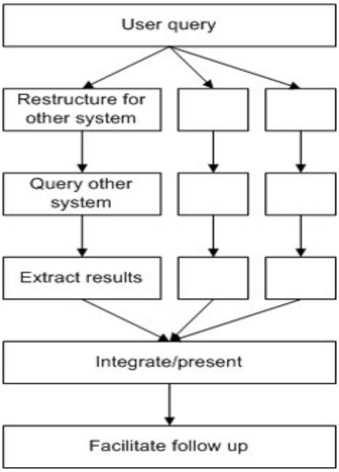


Figure 8.6. The design of BargainFinder (Krulwich 1996)

BargainFinder adopted a typical three-layer design. If we take Figure 8.6 upside down, we find that the bottom layer was a "wrapper," an algorithm that can transform information from a heterogeneous format into a uniform format that can be processed by the upper layer of the agent system (Kushmerick *et al.* 1997). On top of that was an information-processing engine: it classifies, caches, and retrieves information for queries from online shoppers. The outermost layer is a user interface. It provides decision-making functions as well as various visual effects to facilitate comparison-shopping.

The development of BargainFinder did not go through formal SDLC stages. It went directly to system design without requirement analysis because the system was to provide service to the public domain and developers could infer the requirements intuitively. The developers did not built any prototype for testing; instead, a fully functioning agent was delivered at the very beginning. This was because the cost of failure is low for agents compared with iDMSS or other information systems in the organizational domain.

In 1995, the commercialization of the Web was still in its infancy stage. There were relatively few online vendors available, and the product space on the Web was also limited to standardized products like books, CDs, and computer components. The relatively simple Web environment in 1995 made the design and development of BargainFinder considerably easier.

The product space BargainFinder needed to deal with was simple. The data sources BargainFinder targeted were nine online music stores. Music CDs are a fully standardized commodity. They can be uniquely identified via at most two units of information (artist and album). The number of online stores BargainFinder covered was relatively small, which made the critical step of coding a wrapper to retrieve information from each vendor websites technically less challenge. Considering the relatively simple structure of Websites in 1995, the data retrieval and extraction task for BargainFinder was considerably easier than it would be today. Hence, both the data structure of the comparison information (CD title and price) and the number of data sources (nine stores) were simple enough to be handled by BargainFinder.

The legal risk incurred by BargainFinder accessing data in online vendors' site was minimum because the legislature was way behind on new phenomenon like this at that time. Also, BargainFinder was the first-generation comparison-shopping agents, the implications of comparison-shopping on online vendors were not realized yet, even for online vendors. So most online vendors did not know how to deal with this new phenomenon. As a result, only a few of them took any adverse actions (Krulwich 1996).

All the above conditions make the development style of BargainFinder extremely simple and the outcome extremely successfully. It received enormous responses from consumers and media alike. According to Krulwich (1996), within one week of its public release on June 30, 1995, it received an average of 2000 visits per day. BargainFinder set the tone of this new development style and was imitated by its successors.

BargainFinder was a success at that time. However, it could, at most, be considered a simplified demo for future comparison-shopping agents because it was only an experimental software application being put online for testing, and it provided very limited comparison information. Most importantly, though its development style began to depart from the traditional paradigm, thus it still shared many similarities, like both solely depended on technical design to satisfy the user requirements, which proved to be disadvantageous when the Web grew into a highly complex electronic market several years later. BargainFinder completed its mission and soon ceased its operation.

8.2.2 Case Study 2: The Longevity of Pricewatch

Pricewatch was another pioneer comparison-shopping agent that emerged in 1995. Though it provided similar comparison-shopping information to online shoppers as BargainFinder, Pricewatch adopted quite a different development approach.

Unlike the experimental nature of BargainFinder, Pricewatch was engaging in serious online business from the beginning. It positioned itself as a catalog advertiser for partnered online vendors. So the goal of Pricewatch was to officially collect, categorize, post, and promote product information from partnered online vendors on the Web.

This goal let Pricewatch focus more on organizing its vendor partners than how to technically overrun vendors in retrieving information. So instead of developing information-retrieval components like a wrapper and extracting information from vendor Websites, Pricewatch used its proprietary "Info-Link" system to establish an information pipe between itself and the online vendor. The online vendor had to make its information format compatible with the requirements of "Info-Link" and push product information upon request from Pricewatch. Because of this voluntary conformity by online vendors, the information retrieval task became much easier for Pricewatch, which enabled more complex comparison-shopping services for those products that are less standardized than books or CDs.

Though there is limited information about how Pricewatch was developed, based on the explanation on its official websites, we can roughly reconstruct its ISD approach.

The catch of this ad-bot-turned-into intelligent agent was the utilizing of environment change instead of buffering from the environment. The comparison information Pricewatch provided were computer-related products, which were more complex than books and CDs. The online vendors Pricewatch handled were also hundreds in number compared with less than ten for BargainFinder. If we started from the BargainFinder way of thinking and then followed the ISDC paradigm to develop Pricewatch, we would find the job was almost impossible given the technologies (hardware, software, and networking condition) we had back in 1995, not to mention the budget needed for coding labor.

However, Pricewatch cleverly avoided this technical trap by asking the cooperation from online vendors, which worked for small business owners. Because of this change in development thinking in information retrieval, Pricewatch could provide comparison information for more complex products like computer components. It could also retrieve information from far more online vendors than BargainFinder could handle.

This development in thinking also enabled Pricewatch to deal with a more complex environment. For example, vendors partnered with Pricewatch did not have to be online as long as they could feed the product information to the Info-Link system.

We do not know how much time it took for Pricewatch system to be developed, but considering the major challenge was only aggregation of data in same format, it probably took less than BargainFinder needed.

Compared with the brief presence of BargainFinder, Pricewatch has grown steadily through the years and has become one of the major comparison-shopping niche portals in the USA.

8.2.3 Case Study 3: Pricescan.com vs. Shopping.com

Both BargainFinder and Pricewatch have their successors. In 1997, a new comparison-shopping agent called Pricescan was developed using the same approach as BargainFinder. It drew a lot of media attention because of its killer application status at that time. Pricescan achieved a new level of technical sophistication compared with BargainFinder. It could not only aggregate price information from multiple online vendors but also provide nifty features like displaying high, low, and average price trends over the past several weeks for each product upon query.

As an iDMSS, Pricescan could provide more useful assistance in making shopping decisions compared with both BargainFinder and Pricewatch. The comparison information provided by Pricewatch was strictly neutral. According to David Cost, the cofounder, Pricescan did not charge online vendors to be listed in its Website. In addition to bringing the consumer the best price it obtained pricing information not only from vendor web sites but also from offline sources like magazine ads (Anonymous 1998).

However, as we will later demonstrate, the development style of Pricescan, which focused on internal design and realization like BargainFinder did, restrained its development potential and eventually made it a second rate player in the agent competition.

Comparatively, Shopping.com (then named Dealtime.com) was a new generation of comparison-shopping agents. It was among the first group of comparison-shopping agents using traditional marketing efforts to build the concepts of comparison-shopping in consumers (White 2000).

Shopping.com adopted a vendor-partnering development style similar to Pricewatch. However, unlike its taciturn predecessor, it invested intensive efforts in popularizing the concepts of comparison-shopping and brought comparison-shopping into mainstream B2C business[1]. This was partially because comparison-shopping had no off-line counterpart in the pre-Internet age, and many consumers were not accustomed to this shopping behavior, advertising was very important in building the concept, which had been ignored by previous comparison-shopping agents including Pricewatch.

Though this strategy itself seems irrelevant to the development paradigm of the agent, it did have important implications for the thinking for the formulation of a new paradigm. As we repeatedly mentioned, the unpredictability of environment is the major source of risk for ISD. The SDLC paradigm tries to use measures like reducing the development cycle time or balancing the upfront risks to reduce the cost of failure. However, for comparison-shopping agents, these strategies won't work. So they have to invent their own development methodologies. Here, instead of being pressured by the environment, agent service providers conditioned the environment to make it better serve the viability of the agent.

The Web and B2C electronic market have experienced great change since 2000 because of the increase of the number of online shoppers. The comparison-shopping population rose from less than 4% of online shoppers before 2000 to 15%[2] in 2003.

Most significantly, the number of online shoppers using comparison-shopping agents also doubled. The advertisement of new comparison-shopping agents like shopping.com may partially contribute to the increasing popularity of online shopping among ordinary online shoppers.

Shopping.com (renamed from dealtime.com) ranked fourth (behind eBay, Amazon and Yahoo Shopping) among US multicategory e-commerce sites in November 2003, in terms of unique monthly visitors. Most recently, during Mother's Day week in 2004, the number of unique visitors to comparison-shopping site Mysimon.com increased 14% from 274000 to 311000[3]. In its initial public offering (IPO) in October 25, 2004, shopping.com raised $123.7 million.

The increased revenue generated from visitors not only solidified the stability of the internal structure of Shopping.com but also gave it the strength to upgrade its service by merging with and acquiring other comparison-shopping agents, which is another invention of the methodology of this new development paradigm.

---

1 Here we use Media Metrix's 200000 minimum measurement as the benchmark for qualification as a major ecommerce website.

2 Data obtained from: http://www.nielsen-netratings.com/pr/pr_040223_us.pdf

3 Data obtained from: http://www.nielsen-netratings.com/pr/pr_040507.pdf

For example, Shopping.com started with pure price comparison. However, with the increasing number of online vendors available on the Web, identifying quality service became relevant. In addition, when consumers make shopping decisions on unfamiliar products, buyer testimony becomes an important source of information for the decision making. So how to develop the current agent into a new form that could provide these services?

If we follow the thinking of the technical approach like Pricescan, building new system components is the only choice. However, Shopping.com realized this new development cycle via merging and acquisition. It however merged with epinion.com in April 2003 and acquired resellerratings.com in February 2004 to fulfill the new purpose. Epinion.com is an agent specializing in collecting review information on products and resellerratings.com in collecting rating information on online vendors. Both of them are great complementary services for Shopping.com. By integrating these two sources of information, shopping.com became a more powerful comparison-shopping agent that could provide more comprehensive information.

8.3 The Characteristics of the New Paradigm

We summarize the characteristics of this new paradigm reflected in the case studies above into three aspects: start simple in design or by imitation, blend environment solutions to technical challenges, and grow between stasis and punctuation.

8.3.1 Start Simple in Design or by Imitation

An i-DMSS is usually characterized by very complex and sophisticated designs. However, such is not the case for comparison-shopping agents. Comparison-shopping agents not only start simple in design but also show a greater level of universality in their design due to imitation. The simple design is out of three environmental pressures.

First, comparison-shopping is a service for the general public with diversified preferences, so it is very difficult to make a comprehensive user analysis or have specific users involved in the development process. Instead, most of the time, it is the developer using his or her intuition as the user to generate basic requirements. However, the developer is also aware of the uniqueness of individual preference. The design has to be extremely flexible and simple for future change and further development.

Secondly, the general public do not want to spend much time on learning how to use the agent when they use different comparison-shopping agents. According to Zipf's law (Zipf 1949), people always want to minimize their effort as long as they can reach the minimum requirements for the decision quality. Recent controlled experiments also prove that decision makers do not use advanced functions provided by DSS as long as they can meet their minimum decision criteria by using those less advanced functions. This is because using advanced functions requires more effort to learn though it could improve the decision quality (Payne *et al.* 1993, Payne 1982, Todd and Benbasat 1999, Todd 1988). So new comparison-shopping agents have to avoid increasing the learning curve of consumers by providing information in similar formats as other agents, which means a similar interface and format of comparison information, unless the change makes a significant contribution to consumer benefit. This leads to imitation and subsequently to the universality of interface design for comparison-shopping agents.

Thirdly, because of competition from incumbent comparison-shopping agents, new agents usually avoid the risk by adopting similar successful designs of existing agents. This imitation behavior leads to the universality of internal structure and development patterns of comparison-shopping agents.

The universalities of both external and internal design as well as the development pattern simplified the development process at the start.

However, to survive, a comparison-shopping agent must develop itself some competitive advantage in the business process, which can be translated into uniqueness in system- design patterns.

8.3.2 Blend Environmental Solution to Technical Challenges

If we follow the historical evolution of comparison-shopping agents, we will find that the technical challenges of system design are not necessarily overcome through technical evolution or revolution. Instead, major challenges are overcome through reorganizing the relationship with environments. This is a very important feature that differentiates the development paradigm of comparison-shopping agents from those of i-DMSS in a closed domain.

The biggest technical challenge for comparison-shopping agents since 2000 is how to efficiently retrieve information from the Web. Back in 1995, with a relatively simple Web environment and the instant fame of BargainFinder, many similar information-retrieval algorithms were designed and were adopted by subsequent agents to extract information from heterogeneous data sources. These algorithms work perfectly with limited minimum manual maintenance when the product identification information is simple and standardized. As a result, in 1997, more sophisticated BargainFinder-type comparison-shopping agents like pricescan.com and Jango.com began to prosper and dominate the market.

However, situations changed gradually after 2000. When the B2C electronic market became increasingly mature, established brick and mortar players like Wal-Mart joined the online competition together with incumbent online portals like Amazon.com. Tens of millions of small business also put their products online. The product space became exponentially vast and complex. For example, Amazon.com, the biggest online vendor, has an average of 18 million different products online at any moment[4]. Meanwhile, consumers' expectation on the coverage of comparison information had increased substantially.

This increase in expectation was in direct contrast with the constraints on the information-retrieval technique being adopted by most comparison-shopping agents at that time, which still mainly depended on using a wrapper technique to retrieve information from the vendor site directly. The sheer number of online vendors made

---

4 For details see: http://www.ug.it.usyd.edu.au/isys1003/assignments/amazoncase.doc

even minimum manual assistance in information retrieval prohibitively costly. The disadvantage of the BargainFinder-Pricescan structure began to show its limits.

So how to overcome the challenge? Conventional system-development paradigms would pursue more sophisticated techniques, which might have been tried by some established comparison-shopping service providers. However, most agents began to change their relationship with online vendors to overcome the challenge. Specifically, instead of changing their internal design, they changed their relationship with the environment and by changing their relationship with the environment, they not only overcame the technical challenge but also established a stable vendor base. This solution can be traced back to the model used by Pricewatch back in 1995. Interestingly, at that time, Pricewatch adopted this model not because of the technical challenge but because of its original online catalog business model.

Since 2000, many agents changed or strengthened their relationship with online vendors. Instead of retrieving information from online vendors directly, they asked for collaboration from online vendors in comparison information collection. Newly emerged agents like Shopping.com, Pricegrabber.com, *etc.*, began to establish this relationship from the very beginning. The most recent comparison-shopping agents like Pricecomparison.com solicits vendors to join even before their formal launch.

Another technical challenge worth mentioning here is the legal dispute over the access of vendor information, though this is rarely a problem nowadays because more and more vendors realize the benefit of being listed in comparison-shopping. Because some online vendors feel uncomfortable at being price-compared with others, they do not want to be accessed by comparison-shopping agents. Though new techniques like SOAP might still allow agents to avoid a direct block from the vendor side, the legal dispute still looms. In 2002, a Wall Street Journal article named "Are Bots Legal" (Plitch 2002) raised the issue to national attention and gave a final blow to the BargainFinder way of retrieving information online. This provides us another example of how complex information system development could be in an online environment.

8.3.3 Grow Between Stasis and Punctuation

Usually after an i-DMSS in a closed domain has been implemented, the maintenance of the system becomes the main activity, and the system enters into the final stage of its life cycle. There are relatively clear boundaries between system design, development, and maintenance. However, for comparison-shopping agents, it is very difficult to distinguish between the maintenance stage and the development stage.

Because the purpose of comparison-shopping agents is profit seeking, the development pattern of an agent is characterized by a recursive pattern of stasis following by punctuation.

When a comparison-shopping agent is in stasis status, system maintenance is the main routine. The punctuation is usually triggered by environment changes or self-innovation. When the environment of an electronic market changes, comparison-shopping agents have to change their interface or internal structure to accommodate the changed environment accordingly.

For fundamental environment changes like exponentially increasing the number of online vendors and expansion of product space in the last five or six years, these agents have to change their internal structure to survive, like evolving from the development approach adopted by Pricescan.com to those used by Shopping.com. For temporal environment changes like new features adopted by peers, the comparison-shopping agent can adopt similar technical solutions to avoid being left behind. For example, when one or two leading comparison-shopping agents began to include vendor-rating information in their price comparison, almost immediately, other agents began to provide similar services.

Self-innovation is another way that can trigger punctuation. The leading comparison-shopping agents, because they have relatively sufficient R&D support, become the innovation leaders and keep generating new features and providing new services, *e. g.* the merging and acquisition of Shopping.com. By self-innovation, these agents push themselves from stasis into punctuation and then into a new level of stasis.

So, compared with i-DMSS in a closed domain, comparison-shopping agents show a distinctive development pattern.

8.4 Discussion

Apart from the characteristics above, this new paradigm has a major feature: the boundary between the system-development process and the business-expansion process is blurred. The development/design of the agent system can be also interpreted as the development/design of the business model.

Traditionally, SDLC has been characterized by a well-organized rational design and well-controlled implementation though the approaches and methodologies they use vary (Iivari *et al.* 1998). This is mainly due to the environment within which this paradigm is applied: an organization or similar closed domain.

However, as manifested in much field research and case studies, there are several major limitations for this approach. For example, the usage behavior of users in an open domain like the Internet are usually unpredictable; thus, the i-DMSS developed through SDLC may fail to provide adequate support to users because it is not compatible with their mental status. Also, since the environment is in a constantly changing mode, there is not much sense in differentiating between the design and the implementation stages of the system because the system may no longer be applicable to the environment after it is implemented.

In his classic book, Organizations in Action, Thompson (1967) described the major challenge of traditional business as coordinating the conflict between the “rational core” of the organization, which is mainly the technical operation, and the turbulent outside environment. So organizations add a managerial layer outside the rational core to accommodate or buffer changes from their respective environments. Thus, the information system in an organization is behind the business process and plays the role of supporting the business process.

In the case of the comparison-shopping agent, the rational core is completely naked to the environment. The business process for these agents can be translated bit by bit into the technical process. The “rational core” of the agent is no longer buffered by the managerial layer and has to adapt to any change of the environment directly. Instead, managerial activity is used to support the rational core to operate. In other words, the managerial buffering is reduced to development and maintenance of the system within the rational core. This is the fundamental difference for information intermediaries in open domain compared with i-DMSS or other information systems in a closed domain.

8.5 Conclusions and Future Research

This chapter is an exploration study for introducing a new paradigm for the development of comparison-shopping agents as well as similar Web-based i-DMSS. Though we use “new” here, this paradigm is already being widely implemented. However, the theoretical discussion on this paradigm seems to be still not started.

This new paradigm emphasizes the simple launching of the system. It does not distinguish among the design, implementation, and maintenance stages. It alternates the system status between stasis and punctuation.

This paradigm provides the maximum level of flexibility to allow the system to accommodate the complex environment and reduce the risk of failure. It also allows the system to respond rapidly to environment change, thus reducing the cost of delay.

The environment-sensitive nature of this paradigm and its solution style can provide inspirations for the improvement of SDLC especially the conflict between implementation and changing of requirements.

We need considerable further studies to understand the nature and significance of this new paradigm. This chapter can be considered an introductory study in this direction. Future research should focus on more systematic examination of the paradigm following formal methodology.

References

Alavi M and Henderson JC (1981) An evolutionary strategy for implementing a decision support system. Management Science 27(11): 1309-1323

Anonymous (1998) Anneneberg, Bloomberg, and now David Cost and Jeff Trester's PriceScan, proves that price information is power! http://ww1.prweb.com/releases/1998/2/prweb3455.php. Accessed: August 15 2004

Berrisford T, Wetherbe J (1979) Heuristic development: A redesign of system design. MIS Quarterly 3(1): 11-19

Boehm B (1988) A spiral model of software development and enhancement. IEEE Computer 21(5): 61-72

Boehm B, Bose P, Horowitz E, Lee MJ (1995) Software requirements negotiation and renegotiation aids: a theory-W based spiral approach. In Proceedings of 17th International Conference on Software Engineering (ICSE-17), Seattle, WA

Boehm B, Ross R (1988) Theory-W software project management: A case study. In Proceedings of International Conference on Software Engineering, Singapore

Boehm B, Ross R (1989) Theory W software project management: principles and examples. IEEE Transactions on Software Engineering 15(7): 902-916

Bradshaw JM (1997) An introduction to software agents. In: Bradshaw, JM (eds.) Software agents, MIT Press. Cambridge, MA, pp. 3-46.

Henderson-Sellers B, Edwards JM (1993) The fountain model for object-oriented system development. Object Magazine, July-August 1993

Iivari J, Hirschheim R, Klein HK (1998) A paradigmatic analysis contrasting information systems development approaches and methodologies. Information Systems Research 9(2): 164-193

Krulwich B (1996) The bargainfinder agent: comparison price shopping on the Internet. In: Williams, J (eds.) Bots, and other Internet beasties, Macmillan Computer Publishing. Indianapolis, pp. 257-263.

Kushmerick N, Weld DS, Doorenbos RB (1997) Wrapper induction for information extraction. In Proceedings of 1997 International Joint Conference on Artificial Intelligence, Aichi, Japan

Martin J (1991) Rapid application development. Macmillan Coll Div, Indianapolis

Mylopoulos J, Kolp M, Castro J (2001) UML for agent-oriented software development: The tropos proposal. In: Gogolla, M and Kobryn, C (eds.) UML 2001 -- the unified modeling language, LNCS Springer-Verlag. pp. 422-441.

Naumann JD, Jenkins MA (1982) Prototyping: The new paradigm for systems development. MIS Quarterly 6(3): 29-44

Payne JW (1982) Contingent decision behavior. Psychological Bulletin 92(2): 382-402

Payne JW, Bettman JR, Johnson EJ (1993) The adaptive decision maker. Cambridge University Press, Cambridge.

Plitch P (2002) Are bots legal? Wall Street Journal 240(54): R13

Robinson R (1996) Put the Rapid into RAD. Datamation 42(4): 80

Royce W (1970) Managing the development of large software systems: Concepts and techniques. In Proceedings of WESCON, Los Angeles, CA

Sprague RHJ (1980) A framework for the development of decision support systems. MIS Quarterly 4(4): 1-26

Thompson JD (1967) Organizations in action. McGraw-Hill Book Company, New York.

Todd PA, Benbasat I (1999) Evaluating the impact of DSS, cognitive effort, and incentives on strategy selection. Information Systems Research 10(4): 356-374

Todd PA (1988) An experimental investigation of the impact of computer based decision aids on the process of preferential choice (*doctoral dissertation*). The University of British Columbia, Vancouver.

White E (2000) No comparison.Wall Street Journal, October 23, 2000

Wood M, DeLoach SA (2000) An overview of the multiagent systems engineering methodology. In Proceedings of The First International Workshop on Agent-Oriented Software Engineering, Limerick, Ireland

Wooldridge MJ, Jennings NR, Kinny D (2000) The GAIA methodology for agent-oriented analysis and design. Journal of Autonomous Agents and Multi-Agent Systems 3(3): 285-312

Zipf GK (1949) Human behavior and the principle of least effort: An introduction to human ecology. Addison-Wesley Press, Cambridge, MA.

9

A Causal Knowledge-driven Negotiation Mechanism for B2B Electronic Commerce

Kun-Chang Lee[1] and Soonjae Kwon[2]

[1] Professor of MIS, School of Business Administration, Sungkyunkwan University, Seoul 110-745, Korea. e-mail:leekc@skku.ac.kr

[2] Lecturer of MIS, School of Business Administration, Sungkyunkwan University, Seoul 110-745, Korea. e-mail:kwonsj@skku.edu

With the advent of the Internet, the electronic B2B negotiation process has drawn increasing attention from both researchers and practitioners. However, the literature still shows that only structured negotiation terms (SNTs) are being explicitly considered, despite the fact that unstructured negotiation terms (UNTs) should be rendered as well. This chapter proposes a new negotiation support mechanism that can be utilized to incorporate causal relationships between SNTs and UNTs in the process of B2B negotiation, by using a cognitive map. The proposed negotiation mechanism suggests that cognitive maps could be used to represent causal relationships between SNTs and UNTs, both as knowledge representation vehicles and as inference engines. After reviewing the potential of cognitive map in B2B negotiation, we implemented a prototype, CAKES-NEGO, which we then used in illustrative examples in order to examine the validity of our proposed mechanism. The mechanism was tested using two practical scenarios: a structured, twenty-one item questionnaire was developed and applied in order to evaluate the mechanism's validity based on the responses of eleven graduate students. In addition, statistical tests proved that the proposed negotiation mechanism could improve decision performance significantly in B2B negotiations.

9.1 Introduction

With the advent of the Internet, most companies have engaged in at least some form of B2B electronic commerce (Bakos 1998). Unlike B2C, B2B has continued to gain momentum among companies because of its huge potential trade volume and subsequent monetary payoff, its long-term trust, and the necessity of negotiation between trading partners (Dai and Kauffman 2002, Park and Park 2003, Subramaniam and Shaw 2002). Traditional B2B commerce requires the utilization of special negotiation before a deal can be struck between trading partners. Electronic B2B negotiation, however, does not rely upon face-to-face communication, and must therefore foster intelligent decision support from an intelligent negotiation support system. This chapter examines the particular role and potential of cognitive maps in representing the causal relationships among multiple negotiation terms in electronic B2B negotiation.

Factors relating to B2B negotiation can be organized into two groups: "structured" negotiation terms (SNTs) and "unstructured" negotiation terms (UNTs). SNTs are always the primary goal of B2B negotiations because they encompass price, quantity, quality, payment conditions, *etc.* Meanwhile, UNTs have causal relationships with SNTs: resource availability, vendor preferences, labor-management relationships, corporate culture, *etc.* The role of SNTs in the process of B2B negotiation has long been a subject of analysis (Kersten and Noronha 1999, Kersten *et al.* 2003), but the causal relationships with UNTs have been largely ignored. Negotiators should pay due attention to UNTs, because they can have a profound impact on the ultimate quality of SNTs (although one not explicitly addressed in the negotiation process).It is therefore essential to incorporate the causal relationships among SNTs and UNTs into the negotiation process, both objectively and systematically, if the B2B negotiation results are to be mutually beneficial to the negotiation partners. Without resorting to the negotiation support system, however, it is very difficult for decisionmakers to perform B2B negotiation effectively, because the number of SNTs and UNTs to be considered is huge, and the causal relationships that exist among them are very complicated to deal with effectively.

We are proposing a new type of negotiation support system based on cognitive maps as the knowledge representation mechanism and inference engine. The cognitive map, introduced by Tolman (1948) and used later by Axelrod (1976) was originally utilized to represent knowledge in the political and social sciences; that is, to analyze the cause and effect relationships that are perceived to exist among the elements of a given environment. A cognitive map is designed to examine whether the state of one element has an influence on the state of another.

In Figure 9.1, if the market position of the firm improves, then the stock price will increase. This increase in stock price will in turn result in improved credit. From this example, it is easy to see that positive causal links (denoted as '+' in the cognitive map) can be regarded as excitatory relationships, while negative causal links (denoted as '-' in cognitive map) can be regarded as inhibitory relationships between nodes (Zhang *et al.* 1989).The cognitive map thus represents the experts' beliefs and cognition about ill-structured social relationships (Huff 1990), and can offer an interpretation of otherwise complicated geographic information (Liu and Satur 1999).

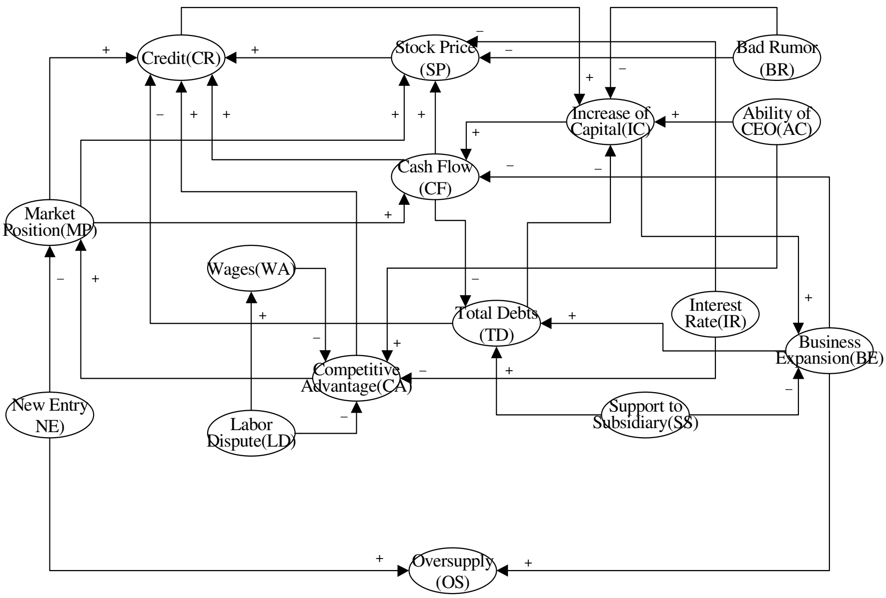


Figure 9.1. Cognitive map for analyzing a firm's credit

Axelrod (1976) used a cognitive map to represent tacit knowledge in the political and social sciences. The cognitive map has also been successfully applied in such areas as decision making in a complex war game (Klein and Cooper 1982), strategic planning (Ramaprasad and Poon 1985), information retrieval (Johnson and Briggs 1994), and distributed decision process modeling (Zhang *et al.* 1994). A cognitive map is composed of nodes, signed directed arcs, and causality values. Nodes represent causal concepts, and signed directed arcs represent the causal relations between two concepts. Causality value is shown by '+' and '-'. Of course, causality value can be "fuzzified" into a real value between –1 and 1 (Lee *et al.* 1992), but Axelrod claimed that simple causality values of '+' and '-' are sufficient for replicating human cognition because decision makers do not typically use a more complicated set of relationships. We adopt the simple cognitive map in this paper in order to show that our cognitive map-driven approach is an effective means of formalizing tacit knowledge about B2B negotiations.

The objective of this chapter, then, is to propose a cognitive map-based negotiation support system for B2B negotiation in which complicated causal relationships among SNTs and UNTs can be represented explicitly via the cognitive map, and inference based causal relationships can be performed given a new B2B negotiation problem. To show the validity of our approach, a prototype, CAKES-NEGO (**CA**usal **K**nowledge-based **E**xpert **S**ystem for **NEGO**tiation), has been implemented. Causal relationships among SNTs and UNTs are represented by the cognitive map, and are stored in it as a kind of causal knowledge in an adjacency matrix. When a new type of B2B negotiation problem is introduced, the causal knowledge in the cognitive map is triggered to produce an appropriate inference result or offer, which is then delivered to the negotiation counterpart. Conversely, when an offer is input into the CAKES-NEGO, a new offer will be created using the stored inferences in the mechanism. CAKES-NEGO continues to be triggered by offers until a final deal is struck between the negotiation counterparts.

9.2 Theoretical Backgrounds

9.2.1 B2B Negotiation

In electronic markets, the so-called EMH or electronic markets hypothesis (Malone *et al.* 1994) is famous for explaining the effects of IT on market structure: when IT is present, more opportunities emerge for market transactions than occur with transactions conducted in a business hierarchy. While there is conflicting evidence about the across-the-board success of EMH (there is no evidence of a shift toward electronic markets in the mortgage-lending industry, for example (Hess and Kermerer 1994)), this can easily be explained by transaction complexity and frequency, buyer/supplier power (Hess and Kermerer 1994), incomplete contracts that cannot capture all the contingencies of the real world (Williamson 1991), and non-contractible investments by suppliers, such as innovation and information sharing (Bakos and Brynjolfsson 1993).

EMH can therefore be expected when a Buyer deals with a small number of suppliers and develops long-term relationships (Bakos and Brynjolfsson 1993), because electronic coordination with fewer suppliers can generate economies of scale (Clemons *et al.* 1993). In a B2B context, academicians and practitioners (Chircu and Kauffman 2000, Barnes-Vieyra and Claycomb 2001) generally accept that industrial business partners have fewer suppliers and longer-term relationships that require less matching, *i.e.*, the "move-to-middle" hypothesis (Clemons *et al.* 1993). This is comparable to B2C electronic commerce, in which a firm operating online has relatively short-term business relationships with a large number of individuals, requiring complicated matching. In the B2C context, operating costs surge as the number of individual customers increases. This is what caused many of the so-called "dot com" companies specializing in B2C to go bankrupt and be expelled from electronic markets in the late 1990s.

After the B2C bubble burst, the B2B sector emerged as the one in which the major impact of electronic commerce is expected. Business Week has released information dealing with changing trends in electronic markets since 1998 (Business Week, 1998 , 1999 , 2000). In its analysis of the B2B market (2000), the B2B sector is estimated as approximately six times larger than the B2C sector, reaching $1.3 trillion by 2003. Three kinds of players exist in the B2B market: a seller market, a buyer market, and a third-party (or "biased") market (Malone *et al.* 1994). Regardless of market type, B2B businesses must still engage in the negotiation process before striking a final deal (Kersten and Noronha 1999). In B2Bs, where multiple seller and buyer companies are trying to make better deals using various types of negotiation strategies, the amount of the transaction in terms of quantity and money is quite big when compared to B2Cs. This causes the negotiation process to become an essential part of B2B transactions (Subramaniam and Shaw 2002, Dai and Kauffman 2002). Certainly, business negotiations are modeled in a form that is suitable for electronic commerce (Kersten and Szpakowics 1998), and web-based negotiation support systems were proposed based on this modeling (Kersten and Noronha 1999).

Multiple firms are involved in B2B negotiation, and the well-known outguessing regress problem (Young 1975) occurs. In the negotiation arena, the outguessing regress problem refers to the fact that no accurate prediction or confident expectation about individual choices can be produced. In order to circumvent the outguessing regress of strategic interactions, rigid assumptions have been adopted, such that (1) the number of players and their identities are assumed to be fixed and known to everyone involved in negotiations, (2) each player knows that the others are rational, and (3) each player's set of negotiation alternatives is fixed and known. Although these rigid assumptions have contributed to producing rigorous theoretic models, they have also made negotiation models specific to certain negotiation cases (Luce and Raiffa 1957, Harsanyi and Selten 1972). B2B negotiation requires more relaxed and realistic assumptions. In line with this need, certain AI models can be viewed as bridges between applications and abstract theoretical models. AI negotiation models have been proven to help players locate an approximate solution strategy, according to bounded rationality principles, by utilizing heuristic search, heuristic evaluation and learning techniques (Rich and Knight 1991). For example, Sycara (1990) proposed a more enriched negotiation model by integrating AI planning, CBR, and other decision-theoretic techniques. An extensive multi-agent negotiation framework has also been developed by several researchers (Kraus and Subrahmanian 1995). In a number of settings, distributed AI models based on multi-agents have been suggested for more robust and effective negotiation models (Sen and Sekaran 1995, Sandholm and Lesser 1995).

Walton and McKersie (1965) have proposed classifying negotiations as integrative and distributive. Distributive negotiation predicts that one party can increase its own value only at the other party's expense, since parties are competitive and claim value. They are interested in achieving more of what is on the table, and are engaged in a fairly simple process of exchanging offers and counter-offers. In contrast, integrative negotiation is based on the premise that solutions can be found, during and because of the process, which reconcile the parties' interests. The key characteristics that distinguish integrative negotiations form distributive are creation of value, focus on interests (and not positions), openness and exchange of relevant information, learning and problem restructuring (Fisher and Ury 1983, Fisher *et al.* 1994, Raiffa 1982, 1996). The process is often complex, as it requires discussion about the parties' interests, the possibilities of expanding the 'pie', and new offers.

These two types of negotiations represent two extremes of a spectrum of mixed negotiations involving a significant element of conflict and a considerable potential for cooperation (Walton and McKersie 1965). Mixed negotiations are more common; negotiators "commit themselves to firm positions (distributive attitude), yet explore options (integrative), make threats (distributive) and yet trust the other negotiator (integrative)" (Fells 1998). In order to build systems capable of conducting and/or supporting mixed negotiations, one needs to understand the requirements for the two extreme types.

The B2B negotiation discussed in this study is most amenable to mixed negotiations because a company engaged in B2B negotiation may want to keep its position (distributive), but still be able to update it in accordance with some information about counter offer or environment (integrative), *etc.*

9.2.2 Cognitive Map and B2B Negotiation

Previously, a cognitive map has proven especially useful in political science (Axelrod, 1976), administrative science (Eden *et al.* 1979), and management science, in which many decision variables and uncontrollable variables are causally interrelated (Eden and Ackermann 1989, Lee and Kim 1997), making it difficult for decision makers to analyze hidden causal relationships that might contribute to their finding more relevant and meaningful solutions (Eden *et al.* 1979, Eden and Jones 1980, Eden and Ackermann 1989, Klein and Cooper 1982, Montazemi and Conrath 1986, Park and Kim 1995, Lee and Kim 1997). For instance, causation in static and dynamic processes was represented by an M-labeled digraph and was used to find solutions for unstructured problems (Burns and Winstead 1985, Burns *et al.* 1989). Information requirement analysis was performed by a cognitive map (Montazemi and Conrath 1986). Kim and Pearl (1987) suggested an inference engine for causal and diagnostic reasoning based on Pearl's (1986) causal network formalism. Eden and Ackermann (1989) proposed SODA (Strategic Options Development and Analysis) that was designed to encourage organizational members to actively define their own strategies. An inference via semantic networking was suggested using binary matrices and matrix multiplication (Burns *et al.* 1989). A cognitive map has been used to represent graph-theoretic behavior to investigate electrical circuits (Styblinski and Meyer 1988), and to describe plant control (Gotoh *et al.* 1989), and an adaptive cognitive map was used to describing virtual worlds (Dickerson and Kosko 1994). Lee *et al.* (1992) developed COCOMAP (Collective Cognitive Modeling) to support group cognitive processes and organizational learning through cognitive modeling. A time variable was introduced into cognitive maps (Park and Kim 1995) so that they could be applied to cases varying with time. Recently, cognitive map has also been used for distributed decision process modeling on networks (Zhang *et al.* 1994), decision analysis (Zhang *et al.*, 1989), stock investment analysis problems (Lee and Kim 1997), and business process redesign (Kwahk and Kim 1999).

B2B negotiation also requires tacit knowledge, since it deals with not only objective and rational SNTs, but also with subjective and firm-specific UNTs. Therefore, in order to accomplish our research premise successfully, we need a rigorous framework for dealing with tacit knowledge about B2B negotiation. Tacit knowledge is usually scattered across all management activities in a given firm, making it very hard to represent explicitly.

Tacit knowledge is often elicited by means of figurative language and symbolism to express the inexpressible (Numata *et al.* 1997). We note that cognitive mapping is well known as a highly-promising technique for capturing tacit knowledge (Lenz and Engledow 1986). Lee and Courtney (1989) have also suggested a cognitive map as a means for constructing organizational memory, and claimed that a cognitive map is superior to common-knowledge representation schemes such as rule and frame. We therefore feel that cognitive map can be used effectively for making tacit knowledge more explicit.

9.2.3 Cognitive Map and Decision-making

Decision-making is an important aspect of management activity (Eierman *et al.* 1995). Some theorists, including Simon (1977), suggest that decision-making is a principle function of organizations. High quality decisions can be expected to lead to more productive actions, quicker problem solving, and better organizational performance. However, decision-making within an organization is not always an easy task, particularly when the underlying problem is complex or poorly structured. Decisionmakers are limited in their cognitive abilities to process complex information (Tayler 1975); they may succumb to a variety of biases (Kahnemann *et al.* 1982), and they may have a difficult time agreeing on a single solution that satisfies differing interests (Zigurs *et al.* 1988). Adding further to the difficulties of decision making is the lack of certainty that a given decision will lead to the desired outcome. As a result of the importance and difficulty of decision making, opportunities presented by computer technology to develop support for decision makers have generated a great deal of interest. Computer programs that feature decision support systems (DSS) have been developed to facilitate the structuring of decisions so that analytical tools, possibly several in combination, can be used to generate solutions (Ariav and Ginzberg 1985). Issues pertaining to the development of computer support for decision makers have also generated a growing body of research, especially since 1975.

Cognitive maps can illustrate causal relationships among the factors describing a given object and/or problem, and they can also describe experts' tacit knowledge about a certain object (Eden 1988, Montazemi and Conrath 1986). Tacit knowledge is personal knowledge embedded in individual experience and, is shared and exchanged through direct face-to-face contact. Tacit knowledge can be communicated in a direct and effective way. The proposed cognitive- map technique has been used to evaluate the factors that affect a given class of decisions, and, most importantly, to enhance the overall understanding within the decision maker's environment. For this reason, cognitive maps are used in our research to analyze and aide decision-making by investigating causal links among relevant domain concepts (Eden and Ackermann 1989, Klein and Cooper 1982).

9.2.4 Inference by Cognitive Map

Usually, a factor is depicted as a node in cognitive map (as shown in Figure 9.1), while a causal relationship between two factors is represented as an edge (or path). Positive causality on an edge from a factor C_i to C_j indicates that increase of C_i causes increase of C_j. Negative causality from C_i to C_j indicates vice versa increase of C_i causes decrease of C_j. To show the inference process using a cognitive map (Figure 9.1), the initial information associated with a new problem is assumed to be collected as follows: no bad rumors (BR), higher market position (MP), more support to subsidiary (SS). In accordance with this information, the inference procedure starts by setting BR to –1, and MP and SS to 1, respectively, with all other nodes set at zero. According to the inference mechanism used by Lee and Kim (1997), the resulting inference history is shown in Table 9.1.

Table 9.1. Inference process by cognitive map

Stage	Factors (Concept Nodes)															
	CR	SP	BR	IC	AC	CF	MP	WA	TD	IR	CA	BE	SS	NE	LD	OS
1	0	0	-1	0	0	0	1	0	0	0	0	0	1	0	0	0
2	1	1	0	1	0	1	0	0	1	0	0	-1	0	0	0	0
3	1	1	0	1	0	1	0	0	-1	0	0	0	0	0	0	-1
4	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0
5	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0

The above results provide the following information:

1) While total debts (TD) increase temporarily because of support to subsidiary (SS), higher market position (MP) leads to increase of cash flow (CF), rise of stock price (SP), and increase of capital (IC) (stage 2).
2) As stock price (SP) rises and cash flow (CF) increases, total debt (TD) decreases and credit (CR) rises (stage 3).
3) With a continuously rising stock price (SP), cash flow (CF), capital (IC), and credit (CR) remain high (stage 4).

Based on the inference process illustrated by the above cognitive map, the company's loan request would probably be accepted because of high credit.

9.3 Experiments Using CAKES-NEGO

9.3.1 Assumptions

CAKES-NEGO is a cognitive-map-based expert system that allows B2B negotiations to be performed seamlessly between buyer and seller.CAKES-NEGO consists of a knowledge base and an inference engine. Figure 9.2 shows the main components of CAKES-NEGO.

Before showing experiments using CAKES-NEGO, it is first necessary to discuss assumptions. We perform experiments from the perspective of a buyer company. For example, the cognitive map in Figure 9.2 depicts the buyer company's tacit knowledge about the B2B negotiation process, where SNTs and UNTs are present and interlinked with each other. SNTs are represented by italics, while the dotted lines denote negative causality, or -1, and the real lines indicate positive causality, or +1. The buyer company in Figure 9.3 is assumed to manufacture a product by purchasing raw materials from suppliers via B2B negotiation on the Internet. Based on the tacit knowledge represented by this cognitive map, the buyer company will negotiate with its counterparts or other possible candidates for a supplier company.

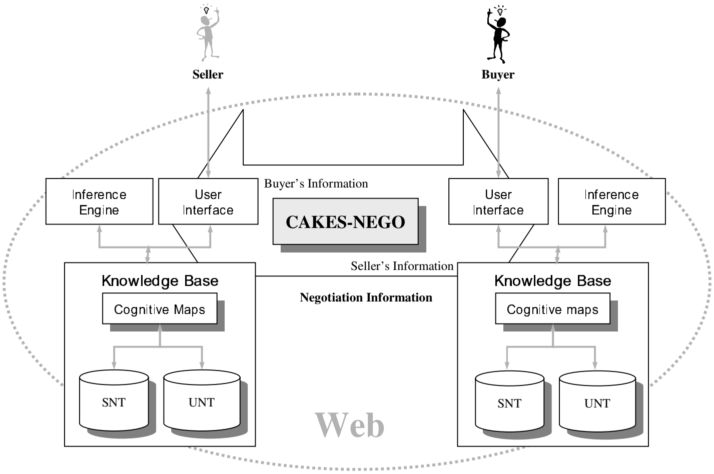


Figure 9.2. Architecture of CAKES-NEGO

Secondly, the cognitive map used in this study has the following preconditions for the sake of making our logic work in a simpler, clearer way:

- the input node always has positive values within an interval [0,1],
- the causality value consists of -1, 0, and +1,
- a 1/2 (or 0.5) threshold value is used to drive the inference process to converge within a finite number of iterations (Kosko 1992, Taber 1991, Wellman 1994).

Thirdly, the cognitive map in Figure 9.3 means that there is an initial version before the B2B negotiation with a possible supplier candidate starts. During the process of B2B negotiation, additional factors or nodes will be added to this cognitive map. We therefore assume that a final cognitive map for the buyer company results as shown in Figure 9.4, based on which the inference will be performed to obtain the final B2B negotiation deal that the buyer company will adopt.

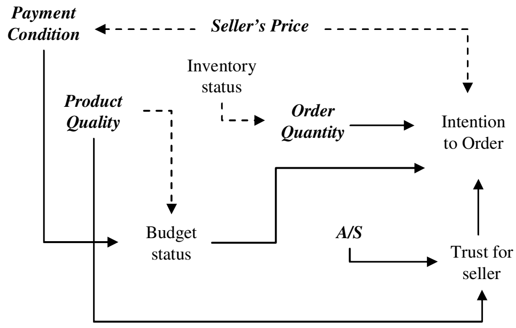


Figure 9.3. Cognitive map for buyer company

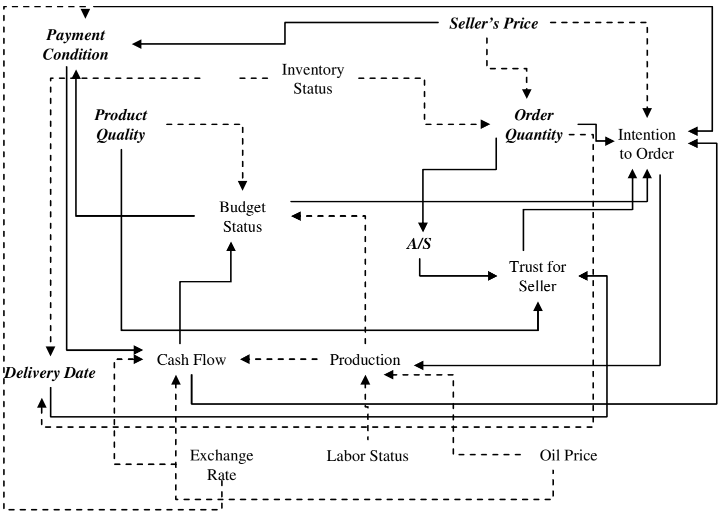


Figure 9.4(a) Final cognitive map for buyer company

9.3.2 Results

In the final cognitive map depicted in Figure 9.4(a), we conclude that the new B2B negotiation problem can be described by the cause-and-effect relationships between six SNTs (SP, OQ, PC, PQ, A/S, DD) and nine UNTs (IS, BS, TS, LS, CF, ER, OP, PR). The abbreviations for each SNT and UNT are explained in Table 9.2. Our goal here is to determine the combination of SNTs that will give the highest intention to order (IO), considering the effects that the UNTs create simultaneously. We will now elaborate on the detailed negotiation terms for each SNT so that we can describe the experiments more practically. Table 9.2 shows node values for SNTs and UNTs from the side of the buyer company. To perform inference (or what-if) analysis with the final cognitive map of Figure 9.4(a), the adjacency matrix <u>**E**</u> is first organized as follows (Figure 9.4(b)).

$$\underline{E} = \begin{array}{cc} & \begin{array}{ccccccccccccccc} \mathrm{PC} & \mathrm{SP} & \mathrm{IS} & \mathrm{OQ} & \mathrm{PQ} & \mathrm{IO} & \mathrm{BS} & \mathrm{TS} & \mathrm{AS} & \mathrm{DD} & \mathrm{CF} & \mathrm{PR} & \mathrm{ER} & \mathrm{LS} & \mathrm{OP} \end{array} \\ \begin{array}{c} \mathrm{PC} \\ \mathrm{SP} \\ \mathrm{IS} \\ \mathrm{OQ} \\ \mathrm{PQ} \\ \mathrm{IO} \\ \mathrm{BS} \\ \mathrm{TS} \\ \mathrm{AS} \\ \mathrm{DD} \\ \mathrm{CF} \\ \mathrm{PR} \\ \mathrm{ER} \\ \mathrm{LS} \\ \mathrm{OP} \end{array} & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Figure 9.4(b) Adjacency matrix of Figure 9.4(a).

Table 9.2(a) shows from SNTs OQ, SP, PC, PQ, A/S and DD, and their corresponding node values that the buyer company thinks reasonable and practical for B2B negotiation. The node values of SP indicate that the buyer company will satisfy more when the seller's prices per unit are lower. However, the node value of OQ is different from that of SP. From the viewpoint of the buyer company, it is preferable to have the order quantity. Outstanding, because then quality control and process control are at a manageable level. The node value of OQ is therefore above 0.9. The greatest node value, of DD, is above 0.85. This occurs when the delivery date is preferable after order placement because it is the tightest when compared to the other delivery dates. Although many scenarios can be derived from the data in Table 9.2, the following scenario is chosen for illustrative purpose.

Table 9.2. Node values for SNTs and UNTs (buyer company)

Criteria	Seller's price (SP)	Order quantity (OQ)	Payment condition (PC)	Product quality (PQ)	After service (A/S)	Delivery date (DD)
Outstanding	above 0.9	above 0.9	above 0.8	above 0.9	above 0.8	above 0.85
Above average	0.7	0.5~0.9	0.6	0.7	0.45~0.8	0.6
Average	0.5	0.5	0.4	0.5	0.45	0.4
Below average	0.3	0.1~0.5	below 0.2	below 0.3	0.2~0.45	below 0.3
Poor	below 0.1	below 0.1	-	-	below 0.2	-

(a) Node values for SNTs

Criteria	Inventory status (IS)	Budget status (BS)	Trust for seller (TS)	Labor status (LS)
Good	above 0.8	above 0.9	above 0.7	above 0.8
Average	0.5	0.6	0.4	0.45
Poor	below 0.2	below 0.4	below 0.2	below 0.2
Criteria	Cash Flow (CF)	Exchange Rate (ER)	Oil Price (OP)	Production (PR)
Increase	above 0.9	below 0.2	below 0.3	above 0.8
Decrease	below 0.6	above 0.8	above 0.7	below 0.3

(b) Node values for UNTs

Scenario:

Let us suppose that the buyer company holds the following node values for SNTs and UNTs, as denoted in Table 9.3.

Table 9.3. SNTs and UNTs for the buyer company

SNTs	Seller's price (SP)	Order quantity (OQ)	Product quality(PQ)	Delivery date(DD)
Criteria	Average	Outstanding	Average	Outstanding
Node Value	0.5	0.95	0.5	0.90
UNTs	Inventory status (IS)	Budget status(BS)	Production(PR)	Labor status(LS)
Criteria	Poor	Good	Good	Good
Node value	0.2	0.9	0.8	0.9

We start our discussion with SNTs. The buyer company is cornered by a delayed production schedule, so that it needs raw materials to be delivered as quickly as possible. We set order quantity as "average", because the order quantity fits with the terms desired (OQ=0.95), and because the seller's price, product quality and A/S are very similar to normal trade terms and conditions of past trade, the buyer company chooses "average" (SP=0.5, PQ=0.5, A/S=0.45). The delivery date is "outstanding" because the product is delivered within three days after purchasing order, satisfying the buyer company's current status: inventory is out of stock (IS=0.2) and delayed production schedule (DD=0.9).

In these circumstances, the buyer company needs to compute the intention to place the order. If the result is higher than the given threshold, the buyer will go ahead with the order. If not, the order will not be placed. Considering the UNTs, inventory status is poor (IS = 0.2), however, the budget status is good (BS = 0.9), production activity is very active (PR = 0.8), and labor status is good (LS=0.9). In these circumstances, the first concept vector $\underline{C}_1$ = (0 0.5 0.2 0.95 0.5 0 0.9 0 0 0.9 0 0.8 0 0.9 0) is created. We perform the cognitive-map-based inference process as follows: the first concept vector, $\underline{C}_1$, is adjusted based on the information in the scenario. Applying a 1/2 threshold for a convergence check, we computed the following inference processes to ensure that convergence can happen within a finite number of iterations.

In Figure 9.5, the cognitive-map-based what-if result is labeled as "inference" and the values inside the bracket can be either 0 or 1. The value will be 1 when the result is greater than the threshold. Otherwise, it will be 0. Inference 1 refers to the result after the first iteration, while inferences 2 and 3 are generated after the second and third iterations.

		PC	SP	IS	OQ	PQ	IO	BS	TS	AS	DD	CF	PR	ER	LS	OP		
$\underline{C}1$	=(	0	0.5	0.2	0.95	0.5	0	0.9	0	0	0.9	0	0.8	0	0.9	0	)	
$\underline{C}1 \times \underline{E}$	=(	1.4	0	0.95	-0.7	0	1.4	-0.3	1.4	0.95	-1.15	0.8	-0.9	0	0	0	)	
->	(	1	0	1	0	0	1	0	1	1	0	1	0	0	0	0	)	-- inference 1
$\underline{C}2$	=(	1	0.5	0.2	0.95	0.5	1	0.9	1	1	0.9	1	0.8	0	0.9	0	)	
$\underline{C}2 \times \underline{E}$	=(	1.4	0	0.95	-0.7	0	4.4	-0.3	2.4	0.95	-1.15	1.8	0.1	0	0	0	)	
->	(	1	0	1	0	0	1	0	1	1	0	1	0	0	0	0	)	-- inference 2
$\underline{C}3$	=(	1	0.5	0.2	0.95	0.5	1	0.9	1	1	0.9	1	0.8	0	0.9	0	)	
$\underline{C}3 \times \underline{E}$	=(	1.4	0	0.95	-0.7	0	4.4	-0.3	2.4	0.95	-1.15	1.8	0.1	0	0.8	0	)	
->	(	1	0	1	0	0	1	0	1	1	0	1	0	0	0	0	)	-- inference 3

Figure 9.5. Scenario 1: inference process by a cognitive map

The equilibrium state is reached when two consecutive inference results are identical. In this scenario, the equilibrium state occurs after two iterations (as can be seen with inference 2 equaling inference 3). By referencing the inference results of $\underline{C}_3 \times \underline{\mathbf{E}}$ in Figure 9.4 and considering both UNTs and SNTs, the buyer company knows that a value of intention to place order (IO) of 4.4 is quite high, because cash flow improves greatly (CF = 1.8). However, the budget status is a little low (BS = -0.3) because a lot of raw materials must be purchased at once. An interesting finding in Scenario is that the buyer company may have a higher intention to order with the UNTs of LS = 0.8 and IS=0.95. This finding has not been addressed or reported in earlier studies of B2B negotiations.

9.4 Discussion

Considering both SNTs and UNTs simultaneously presents a real challenge for B2B negotiators, because it is difficult to compute the chain of influences accurately in light of the large number of cause-effect relationships among them. An incorrect decision may result in a huge loss in B2B revenue; the proposed framework therefore provides a mechanism with which decision makers can consider negotiation terms while making more attractive decisions. To organize this discussion more systematically, we will focus on the evaluation issue.

Nelson *et al.* (2000) proposed seven steps for evaluating cognitive-map-related methodology. Of these, the last step, “validity of findings”, indicates that decision makers should determine whether cognitive map findings actually make sense. In accordance with this criterion, we asked five experts who were originally invited to help us perform member checks (Lincoln and Guba 1985, p. 357) to see whether the results of the two scenarios properly solved the concerns that domain experts usually feel when engaged in the B2B negotiation. Member checks are done by going back to the original expert respondents and asking their opinion about the proposed methodology. The process and purpose of this is to test for factual and interpretive accuracy, and to provide evidence of credibility and trustworthiness similar to internal validity in confirmatory studies (Lincoln and Guba, 1985).

To add more quantitative rigor to the results of the member checks, we also performed a prioritization experiment and Wilcoxon test (Alavi 1982) based on a structured questionnaire survey completed by another group of eleven graduate students engaged in doctoral work in the School of Business Administration. Since decision performance in the proposed framework should be tested statistically, our first job was to configure the questionnaire to evaluate the decision-makers' psychological attitudes towards the framework, as well as its quantitative aspects. Before proceeding further, it seems necessary to consider the meaning of decision performance. Decision performance is "to evaluate the outcomes generated by individuals or groups in accomplishing their task" (Eierman *et al.* 1995). Aldag and Power (1986) also suggested seven constructs of assessing decision makers' attitudes towards decision-making processes: (1) confidence in decision quality; (2) enhancement of problem-solving ability; (3) satisfaction with resource expenditure; (4) perceived acceptability of solution; (5) perceived process structure; (6) perceived process adequacy; and (7) positive affect toward process. Aldag and Power's constructs were adopted to help us organize our questionnaire, which consisted of twenty-one items (see Table 9.5 for details). Valid responses were gathered highlighting the eleven graduate students' attitudes towards solving the two scenarios based on the proposed mechanism.

We compared an experience solving scenarios: "using CAKES-NEGO" and "not using CAKES-NEGO". For this purpose, the eleven graduate students, who were majoring in IS topics, were asked to answer by circling a number from one to seven, arranged horizontally beneath anchor point descriptions "strongly disagree (1)," "neutral (4)," and "strongly agree (7)". Because each respondent was asked to evaluate the questionnaire, either after using CAKES-NEGO or not using CAKES-NEGO, response results were analyzed statistically using the Wilcoxon matched-pairs test (Levin and David 1998). The Wilcoxon test, a nonparametric approach, can be used to compare two variables within one group (Alavi 1982) and has been used broadly in the field of MIS research (Baroudi and Orlikowski 1989). Because the Wilcoxon test does not require a hypothesis about the form of distribution, it was suitable for our case, in which the sample numbered fewer than 30. Table 9.5 shows the Wilcoxon test results, and illustrates that for nineteen of the twenty-one items (not including DP9 and DP11), the subjects perceived a significant difference between the two kinds of experience. The other two items, DP9 and DP11, illustrate that the subjects did not perceive a difference in "implementation of solution" and "frustration" between "using CAKES-NEGO" and "not using CAKES-NEGO" to solve the scenarios.

For "using CAKES-NEGO", all other decision performance measures showed higher means or improved values when compared to "not using CAKES-NEGO", indicating that using CAKES-NEGO to solve the B2B negotiation problem positively influenced the decision-making process, decision outcomes, and decision performance. In a nutshell, Table 9.5 reveals that "using CAKES-NEGO", based on the cognitive map could provide a better decision performance for the two scenarios than "not using CAKES-NEGO".

Table 9.5. Wilcoxon test results

Construct	Items #	Using CAKES-NEGO		Not using CAKES-NEGO		T-value	p-Value
		Mean	SD	Mean	SD		
Confidence in decision quality	My case solution was a good one: DP2	6.18	0.75	5.00	0.45	-2.74	0.01**
	I am not sure my solution was appropriate: DP8 (R)	5.18	0.75	4.27	0.47	-2.64	0.01**
	I am not confident about my solution: DP20 (R)	5.55	0.52	4.27	0.47	-2.89	0.00***
Enhancement of problem-solving ability	Analyzing the case improved my problem-solving skills: DP6	6.18	0.60	4.91	0.54	-2.91	0.00***
	Analyzing the case was a useful learning experience: DP14	6.18	1.08	4.82	0.60	-2.59	0.01**
	I'll be able to handle future problem situations better because of the approach I used to analyze the case: DP19	6.09	0.70	5.00	0.45	-2.76	0.01**
Satisfaction with resource expenditure	It took too much time to solve the case: DP4 (R)	6.00	0.89	4.09	0.94	-2.55	0.01**
	The time and effort used to analyze the case were well spent: DP12 (R)	5.82	0.75	3.91	1.22	-2.83	0.00***
	The approach used to analyze the case wasn't worth the effort: DP18 (R)	5.91	0.54	4.45	0.82	-2.72	0.01**
Perceived acceptability of solution	People in the case who would be affected by my solution would probably be satisfied with it: DP3	5.91	1.04	4.73	0.65	-2.56	0.01**
	I might find it hard to get my solution implemented: DP11 (R)	5.00	0.77	4.45	0.82	-1.40	0.16
	I could easily justify my solution: DP16	6.09	0.83	4.82	0.60	-2.89	0.00***
Perceived process structure	The approach taken to solving the case was very structured: DP1	6.36	0.50	5.09	0.83	-2.56	0.01**
	My analysis of the case was systematic: DP13	6.64	0.67	4.91	0.54	-2.85	0.00***
	I analyzed the case in a step-by-step manner: DP21	6.45	0.52	5.09	0.94	-2.46	0.01**
Perceived process adequacy	I wish I had approached the case differently: DP7 (R)	5.73	0.65	4.36	0.92	-2.55	0.01**
	I really felt lost in trying to tackle the case: DP10 (R)	5.45	0.52	3.91	1.14	-2.70	0.01**
	I may have missed important things in the case: DP15 (R)	5.55	0.52	4.36	1.12	-2.12	0.03*

Positive affect toward the decision process	I am pleased with the approach used to analyze the case: DP5	6.00	0.89	5.27	0.47	-2.13	0.03*
	Analyzing the case frustrated me: DP9 (R)	5.27	0.65	4.64	0.67	-1.73	0.08
	Analyzing the case was interesting: DP17	6.00	1.10	4.73	0.47	-2.36	0.02*

*p<0.05, **p<0.01, ***p<0.001

9.5 Concluding Remarks

B2B negotiation has been dealt with by game theorists, computer scientists, and economists (Byde *et al.* 2002, Conry *et al.* 199, Kauffman and Walden 2001). As the agent technologies have received more attention from researchers and practitioners, B2B negotiation has been tackled from technological agent points of view, as well (Collins *et al.* 2001, 2002, Karageorgos *et al.* 2002). Although these developments have helped B2B negotiation to emerge as an active research area, many IS researchers have neglected to examine B2B negotiation issues from a practical and easy-to-understand viewpoint.

To meet this need, we have proposed using a cognitive map to address the B2B negotiation problem so that a general audience can grasp how B2B negotiations should be handled. Since we have already defined B2B negotiation as a concept encompassing all the activities of electronic commerce between firms on the Internet, it follows that there are a myriad of factors that can potentially affect the performance of B2B negotiations. However, the real problem is how to understand the seemingly complicated relationships among those factors, many of which may be directional or non-directional. Most negotiation models and theories (Fisher and Ury 1991, Mastenbroek 1989, Donnellon 1996) agree that long term cooperation, in a win-win spirit with effective relationship building, is the best option (Ulijn *et al.* 2001), requiring a high degree of involvement in the course of negotiations. Since one of the characteristics of B2B negotiation involves the building of long-term relationships with partners, something that is quite contrary to B2C cases in which short-term relationships are preferable, the argument for the importance of long-term cooperation is also appreciated in B2B negotiation. Therefore, B2B negotiation also requires a high degree of involvement, justifying the use of a cognitive map to formalize tacit knowledge about negotiations and reuse it in new B2B negotiation problems. Without the proposed mechanism, in which tacit knowledge about B2B negotiation is formalized in a cognitive map and used systematically to provide appropriate decision support for new B2B negotiation problems, decision makers cannot be highly involved in B2B negotiation problems.

With the support of two famous B2B companies in South Korea, we profiled several B2B cases and applied our proposed B2B negotiation framework to them. The experimental results showed that the proposed B2B negotiation framework, using a cognitive map, could provide a successful electronic arena in which UNTs and SNTs are integrated and analyzed systematically without neglecting some of them in the course of negotiations. The use of the proposed B2B negotiation framework also implies that decision makers have an overview of the problem at hand, and are able to investigate the possible influences of changes in some UNTs or

SNTs of target variables. In this way, B2B decisionmakers can be more involved in their jobs. We hope that this study arouses further academic interest in cognitive maps in other challenging electronic commerce areas.

Acknowledgements

This paper was supported by a Korea Research Foundation Grant (KRF-2001-041-C00339).

References

Alavi M (1982) An assessment of the concept of decision support systems as viewed by senior-level executives. MIS Quarterly 6:1-9

Aldag RJ, Power DJ (1986) An empirical assessment of computer-assisted decision analysis. Decision Science 17:572 588

Ariav G, Ginzberg MJ (1985) DSS design: a systemic view of decision support Communications of the ACM 28:1045-1052

Axelrod R (1976) Structure of Decision: The Cognitive Maps of Political Elites. Princeton, NJ: Princeton University Press.

Bakos JY, Brynjolfsson E (1993) From vendors to partners: information technology and incomplete contracts in buyer-supplier relationships. Journal of Organizational Computing 3:301-328

Bakos JY (1998) The Emerging Role of Electronic Marketplaces on the Internet. Communications of the ACM 41:35-42

Barnes-Vieyra P, Claycomb C (2001) Business-to-business e-commerce: models and managerial decisions. Business Horizons 44:13-20

Baroudi JJ, Orlikowski WJ (1989) The problem of statistical power in MIS research. MIS Quarterly 13:87-106

Burns JR, Winstead WH (1985) M-labeled digraphs: an aid to the use of structural and simulation models. Management Science 21:343-358

Burns JR, Winstead WH, Haworth DA (1989) Semantic nets as paradigms for both causal and judgmental knowledge representation. IEEE Transactions on Systems Man and Cybernetics 19:58-67

Business Week (1999) A Business-to-Business e-boom. Oct 25:62

Business Week (2000) B2B: the hottest net bet yet ? Jan 17: 36-37

Business Week (1998) How electronic commerce could create the most efficient market of them all: good-bye to fixed pricing ? May 4:71-84

Byde A, Piccinelli G, Lamersdorf W (2002) Automating negotiation over B2B processes. Proceedings of the 13th International Workshop on Database and Expert Systems Applications (DEXA'02) IEEE Computer Society

Chircu AM, Kauffman RJ (2000) Reintermediation strategies in business-to-business electronic commerce. International Journal of Electronic Commerce 4:7-42

Clemons EK, Reddi SP, Row MC (1993) The impact of information technology on the organization of economic activity: the 'move to the middle' hypothesis. Journal of Management Information Systems 10:9-35

Collins J, Bilot C, Gini M, Mobasher B (2001) Decision processes in agent-based automated contracting. IEEE Internet Computing 5:61-72

Collins J, Ketter W, Gini M (2002) A multi-agent negotiation testbed for contracting tasks with temporal and precedence constraints. International Journal of Electronic Commerce 7:35-57

Conry SE, Kuwabara K, Lesser VR, Meyer RA (1991) Multistage negotiation for distributed constraint satisfaction. IEEE Transactions on Systems Man and Cybernetics 21:1462-1477

Dai Q, Kauffman RJ (2002) Business models for internet-based B2B electronic markets. International Journal of Electronic Commerce 6:41-72

Dickerson JA, Kosko B (1994) Adaptive cognitive maps in virtual worlds. International Neural Network Society Annual Meeting World Congress Neural Networks

Donnellon A (1996) Team Talk: The Power of Language in Team Dynamics. Boston, MA: Harvard Business School Press.

Eden C, Jones S (1980) Publish or perish?-A case study. Journal of Operational Research Society 31:131-139

Eden C (1988) Cognitive mapping: a review. European Journal of Operational Research 36:1-13

Eden C, Ackermann F (1989) Strategic options development and analysis (SODA) – using a computer to help with the management of strategic vision. In: Doukidis G, Land F, Miller G. (Eds.) Knowledge-Based Management Support Systems. Ellis Horwood UK, pp 198-207

Eden C, Jones S, Sims D (1979) Thinking in Organizations. Macmillian Press Ltd. London, England

Eierman MA, Niederman F, Adams C (1995) DSS theory: a model of constructs and relationships. Decision Support Systems 14:1-26

Fells RE (1998) Overcoming the dilemmas in Walton and McKersie's mixed bargaining strategy. Industrial Relations, pp 53

Fisher R, Ury W (1983) Getting to Yes Negotiating Agreement Without Giving In. New York Penguin Books

Fisher R, Kopelman E, Schneider AK (1994) Beyond Machiavelli. Tools for Coping with Conflict. Cambridge MA Harvard University Press

Gotoh K, Murakami J, Yamaguchi T, Yamanaka Y (1989) Application of fuzzy cognitive maps to supporting for plant control. SICE Joint Symposium of 15th Systems Symposium and 10th Knowledge Engineering Symposium pp 99-104

Harsanyi JC, Selten R (1972) A generalized Nash solution for two-person bargaining games with incomplete information. Management Science 18:80-106

Hess CM, Kemerer CF (1994) Computerized loan origination systems: an industry case study of the electronic markets hypothesis. MIS Quarterly 18:251-275

Huff AS (1990) Mapping Strategic Thought. Wiley & Sons, New York

Johnson RJ, Briggs RO (1994) A model of cognitive information retrieval for ill-structured managerial problems and its benefits for knowledge acquisition. Proceeding of the 27th Annual Hawaii International Conference on System Sciences, pp 191-200

Kahnemann D, Slovic P, Tversky A (1982) Judgment under Uncertainly: Heuristics and Biases. Cambridge University Press Cambridge England

Karageorgos A, Thompson S, Mehandjiev N (2002) Agent-based system design for B2B electronic commerce. International Journal of Electronic Commerce 7:59-90

Kauffman RJ, Walden EA (2001) Economics and electronic commerce: survey and directions for research. International Journal of Electronic Commerce 5:5-116

Kersten GE, Noronha SJ (1999) WWW-based negotiation support: design implementation and use. Decision Support Systems 25:135-154

Kersten GE, Szpakowicz S (1998) Modeling Business Negotiations for Electronic Commerce. Intelligent Information Systems: Proceedings of the 7th Workshop Warsaw: IPI PAN, 17-28

Kersten MJ, Haley M, Kersten GE (2003) Developing Analytic Cognitive and Linguistic Skills with an Electronic Negotiation System. Proceedings of the Hawaii International Conference on System Sciences 6 9 Big Island Hawaii

Kim JH, Pearl J (1987) CONVINCE: A conversational inference consolidation engine. IEEE Transactions on Systems Man and Cybernetics 17:120-133

Klein JC, Cooper DF (1982) Cognitive maps of decision makers in a complex game. Journal of Operational Research Society 33:63-71

Kosko B (1992) Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence. Prentice-Hall

Kraus S, Subrahmanian VS (1995) Multi-agent reasoning with probability time and beliefs. International Journal of Intelligent Systems 10:459 499

Kwahk KY, Kim YG (1999) Supporting business process redesign using cognitive maps. Decision Support Systems 25:155-178

Lee KC, Kim HS (1997) A Fuzzy Cognitive Map-Based Bi-Directional Inference Mechanism: An Application to Stock Investment Analysis. International Journal of Intelligent Systems in Accounting and Management 6:41-57

Lee S, Courtney JF (1989) Organizational learning system. Proceedings of the Twenty-Second Annual Hawaii International Conference on System Sciences, 492-503

Lee S, Courtney JF, O'Keefe RM (1992) A system for organizational learning using cognitive maps. Omega 20:23-36

Lenz RT, Engledow JL (1986) Environmental analysis: the applicability of current theory. Strategic Management Journal 17:329-346

Levin RI, David SR (1998) Statistics for Management. Seventh Edition, Prentice-Hall

Lincoln Y, Guba EG (1985) Naturalistic Inquiry. Beverly Hills CA: Sage

Liu ZQ, Satur R (1999) Contextual fuzzy cognitive map for decision support in geographic information systems. IEEE Transactions on Fuzzy Systems 7:495-507

Luce RD, Raiffa H (1957) Games and Decisions: Introduction and Critical Survey. New York Wiley

Malone TW, Yates J, Benjamin RI (1994) Electronic markets and electronic hierarchies. In Information Technology and the Corporation of the 1990s, Allen TJ, Scott MS (Eds.) New York: Oxford University Press, pp 61-83

Mastenbroek W (1989) Negotiate. Oxford UK Blackwell

Montazemi AR, Conrath DW (1986) The use of cognitive mapping for information requirements analysis. MIS Quarterly 10:45-56

Nelson KM, Nadkarni S, Narayanan VK, Ghods M (2000) Understanding software operations support expertise: a revealed causal mapping approach. MIS Quarterly 24:475-507

Numata J, Hane K, Lei B, Iwashita Y (1997) Knowledge discovery and sharing in an information system. PICMET '97: Portland International Conference on Management and Technology 713-716

Park JH, Park SC (2003) Agent-based merchandise management in business-to-business electronic commerce. Decision Support Systems 35:311-333

Park KS, Kim SH (1995) Fuzzy cognitive maps considering time relationships. International Journal of Human-Computer Studies 42:157-168

Pearl J (1986) Fusion, propagation, and structuring in belief networks. Artificial Intelligence 29:241-288

Raiffa H (1996) Lectures on Negotiation Analysis. Cambridge MA PON Books

Raiffa H (1982) The Art and Science of Negotiation. Harvard University Press Cambridge MA

Ramaprasad A, Poon E (1985) A computerized interactive technique for mapping influence diagrams (MIND). Strategic Management Journal 6:377-392

Rich E, Knight K (1991) Artificial Intelligence. New York McGraw-Hill Inc

Sandholm TW, Lesser VR (1995) Coalition formation among bounded rational agents. Proceedings of 14th International Joint Conference on Artificial Intelligence (IJCAI)

Sen S, Sekaran M (1995) Multiagent coordination with learning classifier systems. Proceedings of 14th International Joint Conference on Artificial Intelligence (IJCAI)

Simon HA (1977) The New Science of Management Decision Chapter 2, Prentice-Hall Englewood Cliffs NJ

Styblinski MA, Meyer BD (1988) Fuzzy cognitive maps, signal flow graphs, and qualitative circuit analysis. Proceedings of the 2nd IEEE International Conference on Neural Networks (ICNN-87), pp 549-556

Subramaniam C, Shaw MJ (2002) A Study of the Value and Impact of B2B E-Commerce: The Case of Web-Based Procurement. International Journal of Electronic Commerce 6:19-40

Sycara K (1990) Negotiation planning: an AI approach. European Journal of Operational Research 46:216-234

Taber WR (1991) Knowledge Processing with Fuzzy Cognitive Maps. Expert Systems with Applications 2:83-87

Taylor RN (1975) Psychological Determinants of Bounded Rationality: Implications for Decision-Making Strategies. Decision Science 6:409-429

Tolman EC (1948) Cognitive maps in rats and men. Psychological Review 55:189-208

Ulijn JA, Lincke A, Karakaya Y (2001) Non-face-to-face international business negotiation: how is national culture reflected in this medium. IEEE Transactions on Professional Communication 44:126-136

Walton RE, McKersie RB (1965) A Behavioral Theory of Labor Negotiations. New York Graw-Hill

Wellman M (1994) Inference in cognitive maps. Mathematics and computers in simulation 36:137-148

Williamson OE (1991) Comparative economic organization: the analysis of discrete structural alternatives. Administrative Science Quarterly 36:269-296

Young OR (1975) Bargaining: Formal Theories of Negotiations. Illinois: University of Illinois Press

Zhang WR, Chen SS, Bezdek JC (1989) Pool2: A Generic System for Cognitive Map Development and Decision Analysis. IEEE Transactions on Systems Man and Cybernetics 19:31-39

Zhang WR, Wang W, King RS (1994) A-Pool: An Agent-Oriented Open System Shell for Distributed Decision Process Modelling. Journal of Organizational Computing 4:127-154

Zigurs I, Poole MS, DeSanctis GL (1988) A Study of Influence in Computer-Mediated Group Decision-making. MIS Quarterly 13:625-644

10

A Simulation Study of Just-in-time Knowledge Management (JITKM)

Nabie Y. Conteh and Guisseppi A. Forgionne

Department of Information Systems, College of Information Technology and Engineering, University of Maryland, Baltimore County (UMBC), 1000 Hilltop Circle, Baltimore, MD 21250, e-mails: nconteh1@umbc.edu, forgionn@umbc.edu

The problem of "information overload" often results in wasted time and resources and inefficient and unproductive knowledge discovery. In theory, the concept of Just in time knowledge management (JITKM) can help resolve this problem. This chapter tests the theory empirically, and this test supports the theory.

10.1 Introduction

Knowledge creation, access, and use in support of decision making has been facilitated by the emergence and rapid expansion of the Internet and the World Wide Web and by electronic media and libraries (Deagon 1994, Potter *et al.* 1992, King 1990, Shah and Buckner 1991). To be effective, decision makers must have the right information at any given point in time, when it is needed, and in the right amount and form (Conteh and Forgionne 2003a, Conteh and Forgionne 2003b). Such knowledge provision can be called Just-in-time knowledge management (JITKM).

This chapter presents the JITKM concept and demonstrates the benefits for decision-making support. First, there is an overview of the literature in relation to the problems of knowledge delivery. Next, the chapter discusses JITKM characteristics and the role of the concept in overcoming knowledge-base difficulties. Then, there is an empirical analysis of the JITKM concept. Finally, there is a discussion of the study's implications and a presentation of recommendations for maximizing the benefits derived from the use of knowledge bases.

10.2 The Role and Challenges of Domain Knowledge in Organizations

Firms like KPMG, Buckman Laboratories, Andersen Consulting and AMS (Alavi 1997; Power and Karparthi 1998, Klein and Methlie 1995, Forgionne 1999) have practiced experience and knowledge sharing successfully (Alavi 1997, Power and Karparthi 1998, Conteh and Forgionne 2005b, Forgionne 1999). Yet, the process creates challenges. The knowledge manager must identify, from large volumes of information, useful knowledge (Alavi 1997). Once identified, theinformation must be contextualized to become easily searchable and readily available when and how needed. Another challenge is to get employees to use knowledge management tools, which can be perceived as burdensome and time ineffective (Power and Karparthi 1998, Forgionne 1999). Sharing the stored knowledge can be time consuming and frustrating (Conteh and Forgionne 2005b).

10.3 Knowledge and Decision-making

Knowledge can have a profound effect on the professional's decision-making (Dean and Sharfman 1996). For the decision making to be effective and efficient, knowledge will be needed in a timely manner at each step and phase of the process. General, or even expert, knowledge will not be as useful as knowledge that is focused and pertinent to the decision task. For example, a model that precisely and explicitly relates criteria to alternatives and events, even in a nonquantitative manner, will be more useful than a general statement of the relationships involved in the problem.

Just-in-time knowledge management (JITKM) ensures that a person or group performing a specific task related to an overall work process readily receives whatever knowledge he or she needs just when it is needed (Conteh and Forgionne, 2003b; Conteh and Forgionne 2005b). As a result, JITKM can incrementally reduce task lead time and facilitate a seamless work flow. It not only strives against long lead times by pre-empting delays and chaos associated with information overload but also saves money linked with the storing of data and labor in handling knowledge and work-in-process inventories (Conteh and Forgionne 2004a).

10.4 Intelligent Decision Support Systems (IDSS)

A number of information systems exist to generate knowledge for decision-making support. Collectively, these systems can be called intelligent decision support systems (IDSS) (Hans and Peter 1992). These systems integrate the functions of decision support systems (DSS) and knowledge based systems (KBS) to assist decision makers in building analytical models, offer advice on specific problems tasks, assist decision makers in performing decision analysis, and explain conclusions and recommendations (Hunsaker and Hunsaker, 1981; Tansley and Hayball, 1993; Sensiper S. *et al.* 1998; Silverman, 1994).

Usually, the support is offered in a fragmented and incomplete manner with the focus on general problem knowledge and specific advice as viewed from a narrow perspective. In short, traditional IDSS has not provided JITKM. However, the integration of a JITKM capability within DSS can enhance the quality and efficiency of the decision-making support, create synergistic effects, and augment decision-making performance and value (Silverman 1994, Shim *et al.* 2002, Fulmer 1999, Shah and Buckner 1991).

This theory suggests the following research question: Can a JITKM-enhanced DSS improve decision-making? The null hypothesis is that a JITKM-enhanced DSS will result in no improvement in decision-making when compared to a traditional DSS. The alternative hypothesis would be that a JITKM-enhanced DSS would improve decision making when compared to a traditional DSS. To answer this question, we utilized a semistructured decision situation to collect data and test the hypotheses.

10.5 Decision Situation

The decision situation involves a market in which an organization competes for a product's four-quarter total market potential on the basis of price, marketing, and other factors. The demand for the organization's software products will be influenced by, (1) its actions, (2) the competitors' behavior, and (3) the economic environment (Klein and Methlie 1995).

In this situation, decision makers will focus on the key uncontrollable events – competitors' marketing and price, the seasonal index, and the economic index – and the major controllable actions – price, marketing, research and development, and production.

10.5.1 Problem Scenario

A model of the simulated organization is delivered through a software package called AIS (academic information systems) (McLead 1986). A large group of simulated decision makers utilized this software to generate the input variables for the models. Simulated behavior was generated from theoretical and empirical research results reported in previous studies. Since the simulated users were given results from the inputs and could alter their behavior accordingly, this support was termed basic DSS processing.

In contrast, the intelligent just-in-time decision support system (IJDSS) offered advice on the input values to the simulated subjects. We assumed that some, but not all, of the subjects would accept the advice. Subjects not accepting the advice, either partially or completely, would get the same outcomes from the IJDSS processing as would occur in the DSS processing. Hence, differences in decision outcomes could be attributed solely to the subjects' input values for the controllable and uncontrollable variables.

10.6 The Intelligent Just-in-time Decision Support System (IJDSS)

IJDSS advice was rendered through an intelligent IJDSS. Domain and technical expertise is delivered dynamically the IJDSS, thereby making the system intelligent. Desirable input values can be derived from the relationships provided in the AIS software manual. The values generate good though not necessarily maximum profits.

These desirable values were stored in a knowledge base in the IJDSS. Users seeking advice would trigger an intelligent agent that would access the knowledge base, retrieve the suggested values, and display the suggestions to the subject. If the user accepted the advice, the agent would attach the suggested input values to the simulation model, and calculate the corresponding profits. The agent would also record whether the advice was accepted and assign the record to the corresponding subject.

The same numbers of observations were generated for the IJDSS processing as were generated from the DSS processing. Comparisons between the two systems (DSS and IJDSS) were described and used to address the main research issue in this study.

10.7 Summary of Results

The main research issue in this study is to determine if an IJDSS could improve decision making. Improvement can be measured in terms of the outcome and process of decision making.

10.7.1 Outcome Test

The outcome was measured by the organization's profits. A t-test was used to evaluate the mean profits from the use of the decision support system and the Intelligent just-in-time decision support system. The test results, which are summarized in Exhibit 10.1, indicated that there was a statistically significant difference between the mean profits between the simulated users of each system. These results, and the corresponding means, indicate that IJDSS use led to higher mean profits than DSS use. Put another way, the IJDSS improved the outcome from decision making.

10.7.2 Process Tests

Outcome occurs through the process of decision-making (Davenport and Hansen, 1998). This process can be characterized as intelligence, followed by design and choice, and concluded with implementation (Forgionne 2002, Turban 1993).

Exhibit 10.1. Two Sample t-tests for the means of PBT and PBTJ 10 000 subjects

Sample statistics

Group	N	Mean	Std. Dev.	Std. Error
PBT	4000	-7952299	3.31E6	52337
PBTJ	4000	55882974	4.87E7	769843

Hypothesis test

Null hypothesis: Mean 1 - Mean 2 = 0

Alternative: Mean 1 - Mean 2 ^= 0

If variances are	t statistic	Df	Pr > t
Equal	-82.729	7998	<0.0001
Not equal	-82.729	4036.0	<0.0001

In this application, intelligence and early design have been completed prior to system use and are represented by the simulation model delivered by both the DSS and IJDSS. Later design (operationalizing the formulated model) is achieved by assigning values for the uncontrollable events (competitor's price, competitor's marketing, the seasonal index, and the economic index) in the simulation model. Choice is achieved by assigning the decision variables (price, marketing, research and development, and plant investment) in the simulation model. Implementation involves the calculation of profits from the assigned decision and uncontrollable input values. In the DSS, the input values are generated by the subjects through the simulation process. In the IJDSS, the values are assigned by the advice delivered through the intelligent agent (for subjects accepting the advice). Hence, process improvement can be measured by the differences in the controllable and uncontrollable input values between the DSS and IJDSS. If there are no differences, advice was not taken, and there was no process improvement that could be attributed to the IJDSS.

Sample t-test were conducted comparing the values of the uncontrollable variables for the DSS and the IJDSS, respectively: DSS economic index (E), IJDSS economic index (EJ), DSS seasonal index (SI), IJDSS seasonal index (SIJ), DSS competitor marketing (CM), IJDSS competitor marketing (CMJ), the DSS competitor price (CP), and the IJDSS competitor price (CPJ). The results are reported in Exhibits 10.2 -10.5. The results from Exhibits 10.2 -10.5 indicate that there were no statistically significant differences between the values for the uncontrollable inputs in the DSS and IJDSS. These results indicate that the IJDSS did not improve the design phase of decision-making.

Exhibit 10.2. Two sample t-tests for the means of E and EJ 10 000 subjects

Sample statistics

Group	N	Mean	Std. Dev.	Std. Error
E	4000	1.882129	0.4683	0.0074
EJ	4000	1.889703	0.4737	0.0075

Hypothesis test

Null hypothesis: Mean 1 - Mean 2 = 0
Alternative: Mean 1 - Mean 2 ^= 0

If variances are	t statistic	Df	Pr > t
Equal	-0.719	7998	0.4720
Not Equal	-0.719	7997.0	0.4720

Exhibit 10.3. Two sample t-tests for the means of SI and SIJ 10 000 subjects

Sample statistics

Group	N	Mean	Std. Dev.	Std. Error
SI	4000	1.828304	0.597	0.0094
SIJ	4000	1.825968	0.5939	0.0094

Hypothesis test

Null hypothesis: Mean 1 - Mean 2 = 0
Alternative: Mean 1 - Mean 2 ^= 0

If variances are	t statistic	Df	Pr > t
Equal	0.175	7998	0.8607
Not equal	0.175	7997.8	0.8607

Exhibit 10.4. Two sample t-tests for the Means of CM and CMJ 10 000 subjects

Sample statistics

Group	N	Mean	Std. Dev.	Std. Error
CM	4000	2497278	3.09E6	48792
CMJ	4000	2499068	3.09E6	48834

Hypothesis Test

Null hypothesis: Mean 1 - Mean 2 = 0
Alternative: Mean 1 - Mean 2 ^= 0

If Variances Are	t statistic	Df	Pr > t
Equal	-0.026	7998	0.9793
Not equal	-0.026	7998.0	0.9793

Exhibit 10.5. Two sample t-tests for the means of CP and CPJ 10 000 subjects

Sample statistics

Group	N	Mean	Std. Dev.	Std. Error
CP	4000	175.1487	63.505	1.0041
CPJ	4000	175.4892	63.441	1.0031

Hypothesis test

Null hypothesis: Mean 1 - Mean 2 = 0
Alternative: Mean 1 - Mean 2 ^= 0

If variances rre	t statistic	Df	Pr > t
Equal	-0.240	7998	0.8104
Not equal	-0.240	7998.0	0.8104

Similar tests were conducted comparing the values of the controllable variables for the DSS and the IJDSS respectively: DSS Price (P), IJDSS Price (PJ), DSS marketing (M), IJDSS marketing (MIJ), DSS plant investment (PI), IJDSS plant investment (PIJ), the DSS research and sevelopment (RD), and the IJDSS research and development (RDJ). The results are reported in Exhibits 10.6 - 10.9.

Exhibit 10.6. Two sample t-tests for the means of P and PJ 10 000 subjects

Sample atatistics

Group	N	Mean	Std. Dev.	Std. Error
P	4000	166.4833	34.339	0.543
PJ	4000	47.80504	6.4576	0.1021

Hypothesis Test

Null hypothesis: Mean 1 - Mean 2 = 0
Alternative: Mean 1 - Mean 2 ^= 0

If variances are	t statistic	Df	Pr > t
Equal	214.815	7998	<0.0001
Not equal	214.815	4281.5	<0.0001

Exhibit 10.7. Two sample t-tests for the means of M and MJ 10 000sSubjects

Sample statistics

Group	N	Mean	Std. Dev.	Std. Error
M	4000	729364.8	123162	1947.4
MJ	4000	270589.6	21287	336.57

Hypothesis test

Null hypothesis: Mean 1 - Mean 2 = 0
Alternative: Mean 1 - Mean 2 ^= 0

If variances are	t statistic	Df	Pr > t
Equal	232.146	7998	<0.0001
Not equal	232.146	4237.7	<0.0001

Exhibit 10.8. Two sample t-tests for the means of RD and RDJ 10 000 subjects

Sample statistics

Group	N	Mean	Std. Dev.	Std. Error
RD	4000	688693.6	169784	2684.5
RDJ	4000	115565.3	200564	3171.2

Hypothesis test

Null hypothesis: Mean 1 - Mean 2 = 0
Alternative: Mean 1 - Mean 2 ^= 0

If variances are	t statistic	Df	Pr > t
Equal	137.940	7998	<0.0001
Not equal	137.940	7785.8	<0.0001

Exhibit 10.9. Two sample t-tests for the means of PI and PIJ 10 000 subjects

Sample statistics

Group	N	Mean	Std. Dev.	Std. Error
PI	4000	7025740	9.35E6	147904
PIJ	4000	2446849	3.31E6	52340

Hypothesis test

Null hypothesis: Mean 1 - Mean 2 = 0
Alternative: Mean 1 - Mean 2 ^= 0

If variances are	t statistic	Df	Pr > t
Equal	29.185	7998	<0.0001
Not equal	29.185	4985.1	<0.0001

The results from Exhibits 10.6-10.9 indicate that there were statistically significant differences between the values for the controllable inputs in the DSS and IJDSS. These results indicate that the IJDSS did improve the choice phase of decision-making.

10.7.3 Canonical Correlations

Collectively, the sample t-tests suggest that the IJDSS improved the outcome from decision making by enhancing the choice phase in the process. To further test this interrelationship, we conducted a canonical correlation of DSS controllable and uncontrollable inputs against IJDSS controllable and uncontrollable inputs. The main results are summarized in Exhibit 10.10.

Exhibit 10.10's results indicate that DSS and IJDSS uncontrollable inputs are significantly correlated, but IJDSS controllable inputs are not significantly correlated with DSS controllable inputs. These results suggest that the controllable inputs in the IJDSS are statistically significantly different from the corresponding inputs in the DSS.

Exhibit 10.10. Canonical correlation of DSS and IJDSS inputs

	Approximate Likelihood Ratio	F Value	Num DF	Den DF	Pr > F
1	0.35038057	16.46	32	764	<0.0001
2	0.69769423	11.06	15	383	<0.0001

To determine if outcomes collectively were related to process collectively, we conducted a canonical correlation of profits against the controllable and uncontrollable inputs. The results are reported in Exhibit 10.11.

Exhibit 10.11. Canonical correlation of DSS and IJDSS profits against inputs

	Likelihood Ratio	Approximate F Value	Num DF	Den DF	Pr > F
1	0.00003232	173.73	64	2221.4	<0.0001
2	0.02763048	41.09	49	1959	<0.0001
3	0.29734022	15.00	36	1697.8	<0.0001
4	0.68290958	6.22	25	1439.1	<0.0001
5	0.92148213	2.01	16	1186	0.0102
6	0.98834008	0.51	9	946.87	0.8693
7	0.99744612	0.25	4	780	0.9100
8	0.99996665	0.01	1	391	0.9091

The results in Exhibit 10.11 indicate that profits are significantly correlated with the controllable inputs but not with most uncontrollable inputs. These results suggest that the controllable inputs are the main contributors to profits.

These two canonical correlations confirm the previous t-test findings. Namely, the IJDSS improved the outcome from decision making by enhancing the choice phase in the process.

10.8 Conclusions and Implications

The broad conclusion from the conducted simulation study is that the IJDSS, relative to the DSS, helps improve the process of and outcome from decision making. Moreover, from the hypotheses tested, it can be inferred that the input of the right values for the controllable variables, which in essence constitute the process steps, led to the improved profit outcomes.

There are some limits on these conclusions. For one thing, the IJDSS advice is predetermined by the designer (AIS manual), which may be different in other future circumstances. Also, the market simulation used was for only 1 year (4 quarters), which makes it difficult to draw long-range conclusions. In addition, the categories of users or decision styles used are arbitrary though based on empirical and theoretical assumptions. Finally, the input values ranges are arbitrary though based on empirical and theoretical assumptions.

Nevertheless, the results of this study confirm previous research that intelligent decision-making support systems can significantly improve both the outcomes from, and process of, strategic decision making. Unlike previous research, this study offers a different approach to intelligent decision-making support based on the IJDSS concept and the nonsubject simulation approach.

The results imply that the just-in-time concept is superior to traditional knowledge and expert system approaches in guiding the decision maker toward an effective policy or strategy. Another implication is that intelligent agents can offer an effective mechanism to deliver just-in-time knowledge. Such an implication could potentially extend the use of agents in intelligent decision-making support systems beyond simple processing efficiency toward true and direct decision process support. In this study, the agent processing was delivered through the simulation model. Such delivery implies that model relationships, whether in quantitative or qualitative form, can be effective agents in intelligent decision-making support systems. Moreover, this form of agent delivery can offer goal seeking and sensitivity analyses automatically to system users.

References

Alavi M (1997). KPMG Peat Marwick U.S.: One Giant Brain. Boston: Harvard Business School, Case 9: 397-108.

Deagon B, (1994) Has Your firm Changed Ways to Compete in a Knowledge Economy ? Investor's Business Daily, pp 3.

Conteh NY, Forgionne GA (2003a) Intelligent decision-making support, through Just-In-Time Knowledge Management, In: Proceedings of the 7th International Conference on Knowledge-Based Intelligent Information and Engineering Systems, Oxford University, UK, September 3-5, pp. 101–106.

Conteh NY, Forgionne GA (2003b) Delivering enhanced decision-making support through Intelligent Just-In-Time Decision Support Systems, In: Proceeding of the 34th annual conference of the Decision Sciences Institute, Washington DC, November 22-25, pp. 1–6.

Conteh NY, Forgionne G (2004a) Simulation modeling as an information systems evaluation tool, In: Proceeding of the 35th annual conference of the Decision Sciences Institute, Boston, MA, November 20-23, pp.1501-1506.

Conteh NY, Forgionne GA (2005a) The design of dynamic just-in-time agent technology through simulation modeling, In: Frontiers in Artificial Intelligence and Applications Knowledge-Based Intelligent Engineering Systems, IOS Press B.V., Amsterdam, Netherlands, pp 91-101

Conteh NY, Forgionne GA (2005b forthcoming) A just in time approach to intelligent decision support systems Accepted in In: Alimi A (ed) New Trends in the Design of Intelligent Decision Systems: Special issue of the Journal of Decision Systems.

Davenport, TH, Hansen MT (1998) Knowledge Management at Andersen Consulting., Boston: Harvard Business School, Case 9: 499-032.

Dean JW, Sharfman MP (1996) Does decision process matter? a study of decision-making effectiveness. Academy of Management Journal, 39(2): 368-396.

Forgionne GA (2002) Decision-making support systems architecture. In: Forgionne GA, Gupta JND, Mora M (eds.) Decision-making Systems: Achievements and Challenges for the New Decade. Idea Group Publishing, Hershey PA, pp 1-19.

Forgionne GA (1999) An AHP model of DSS effectiveness, European Journal of Information Systems, 8(2): 95-106.

Fulmer WE (1999) Buckman Laboratories, Boston: Harvard Business School, Case 9-800-160.

Hans W G, Peter W (1992) Intelligent decision support systems., Decision Support Systems, 8(4): 317-332

Hunsakers PL, Hunsaker JS (1981) Decision styles – in theory, in practice Organizational Dynamics, 10: 23–36.

King R (1990) DSS: Complementing the Small Business Manager, In: with John M, Robert R (eds) Proceedings of the Midwest Decision Sciences Institute, May 2-4, pp 168-170.

Klein MR, Methlie LB (1995) Knowledge-based Decision Support Systems – with applications in Business, New York: John Wiley & Sons.

McLeod RP (1986) Academic Information Systems (AIS) Software Package II, available from R. P. McLeod, 1106 Glade, College Station, TX 77840.

Potter W, Byrd T, Mille J, Kochut K (1992) Extending decision support systems: the integration of data, knowledge, and model management. Annals of Operations Research, 38: 501-527.

Power DJ, Karparthi S (1998) The changing technological context of decision support systems In: Berkeley D, Widmeyer G, Brezillion P, Rajkovic V (Eds.) Context-Sensitive Decision Support Systems. London, Chapman and Hall.

Sensiper S (1998). American Management Systems: The Knowledge Centers. Harvard Business School, Case 9-697-068.

Shah V, Buckner G. (1991) Intelligent decision support systems. The Journal of Computer information Systems, 31(2): 61-65.

Shim JP, Warkentin M, Courtney JF, Power DJ, Sharda R, Carlsson C (2002) Past, present, and future of decision support technology. Decision Support Systems, 33(2): 111-126.

Silverman BG (1994) Unifying expert systems and the decision sciences., Operations Research, 42(3): 393-413

Tansley DSW, Hayball CC (1993) KBS Analysis and Design, A KADS Developer's Handbook Prentice Hall.

Turban E (1993) Decision Support and Expert Systems: Management Support Systems, Macmillan Publishing Company.

11

An IDSS for Regional Aquaculture Planning

Omar F. El-Gayar and Kanchana Tandekar

College of Business and Information Systems,
Dakota State University, Madison, South Dakota, USA,
e-mails: Omar.el-gayar@dsu.edu, tandek@pluto.dsu.edu

The field of regional planning is characterized by the large number of issues and attributes involved, and regional planning for aquaculture development is no exception. Moreover, aquacultural plans do not have clearly defined objectives and require information that, if it exists, is often imprecise and uncertain. This chapter presents the development of an intelligent decision support system (IDSS) for regional aquaculture planning. The IDSS applies fuzzy set theory to multiple criteria decision-making (MCDM) in regional aquaculture planning. A case study from Egypt demonstrates the proposed IDSS and the fuzzy MCDM framework.

11.1 Introduction

The aquaculture sector currently produces 36 per cent of the world's fish supply, up from 7 per cent in 1970. Moreover, according to Jiansan Jia, Chief of FAO's Inland Waters and Aquaculture Service, aquaculture's contribution to feeding the hungry will become increasingly important in years to come. In fact, some projections suggest that captures by traditional wild fisheries will stagnate within the next 30 years; According to Jiansan Jia, "*Aquaculture is really the only way to meet the gap between supply and growing world demand for fish to eat.*" (FAO, 2003).

However, aquaculture systems are biocomplex systems involving a dynamic web of interrelationships that arise when biological, physical, chemical, and human components interact. The complexity of such systems poses significant challenges when planning for regional aquaculture development, most notable are:

- Planning and managing for sustainability
- Accommodating multiple development goals
- Accommodating uncertainty

According to the final report of the World Commission on the Environment and Development (also known as the Brundtland report) (WCED, 1987), sustainable development (SD) is defined as "*development that meets the needs of the present without compromising the ability of future generations to meet their own needs*". As noted by Andriantiatsaholiniaina and Phillis (2000) one of the most challenging questions to address in the new millennium is how to assess, build, and sustain economies that allow us to enjoy a sufficiently high standard of living without destroying our natural and biological support. In effect, sustainability implies an ongoing dynamic development, driven by human expectations about future opportunities and is based on present economic, ecological and societal issues.

As such, sustainability is inherently a multcriteria concept and any decision support system assessing the sustainability of a system needs to employ techniques suitable for making decisions in the presence of multiple, usually conflicting criteria. Such techniques are referred to as multcriteria decision-making techniques (MCDM). While MCDM has been applied extensively in various application domains, their application in the management of biocomplexity in general and aquaculture in particular is limited.

Moreover, while there is a growing consensus regarding the implications of sustainability, the complexity and vagueness of sustainable development (SD) render it difficult to define or measure. As such, soft computing techniques such as fuzzy logic with their ability to handle natural phenomena that rarely have crisp boundaries and therefore cannot be approached in the classical way with binary logic (fragile or not: 0 or 1) are particularly attractive.

According to Carlsson (1999) the theory of fuzzy logic provides a good mathematical and methodological basis for capturing the uncertainties associated with human cognitive processes. In effect, fuzzy logic provides the means for representing vagueness and incompletely understood concepts. Moreover, Cox (1995) identifies specific capabilities for fuzzy systems making such systems a powerful tool in the design and construction of modern intelligent decision support systems, and particularly suited for modeling and managing biocomplex systems. These capabilities include reduced cognitive dissonance, ability to handle multiple conflicting, cooperating, and collaborating experts, improved knowledge representation, improved and more powerful uncertainty calculus for handling noisy or imprecise data, reduced rule set, ability to handle high complex nonlinear problems, and reduced development and validation times. In effect, fuzzy systems offer the high-level flexibility and knowledge representation of conventional decision support and expert systems combined with the power and analytical depth of natural computing paradigms such as neural networks and genetic algorithms (Cox 1995).

In summary, MCDM techniques and fuzzy logic directly contribute towards addressing the aforementioned challenges in planning that involves biocomplex systems such as planning for aquaculture developments. Specifically:

- MCDM techniques support decision making in the presence of multiple, often conflicting criteria encountered when planning for sustainability and thereby allow for the analysis of tradeoffs between environmental and economic objectives as well as system optimality and reliability.
- Fuzzy models provide a better, more natural knowledge representation, with the capabilities for handling uncertain and vague information, encoding common sense and expert knowledge, its interpretability and tractability, and its low cost in design and maintenance.

- Fuzzy sets can mitigate some of the limitations inherent in MCDM techniques, such as the treatment of noncommensurate units, the ranking procedure for a solution, and the degree of discrimination of attribute values.

This chapter presents the development of an intelligent decision support system (IDSS) for regional aquaculture planning. The IDSS applies fuzzy set theory to multicriteria decision making (MCDM) in aquaculture planning. The aim is to address the challenges facing planners and decision makers when planning for regional aquaculture development, most notable is the multiplicity of developmental criteria and the inherent uncertainty encountered when planning for aquaculture development.

The chapter is organized as follows; the next section presents a brief review of the fuzzy multicriteria decision-making literature with an emphasis on applications in biocomplex systems. The following section presents the development of the IDSS followed by a demonstration of the applicability of the proposed IDSS using a case study from Egypt. The final section concludes the chapter.

11.2 Related Work

11.2.1 IDSS for Regional Aquaculture Development

In a survey of decision support system applications (1988-1994), Eom *et. al.* (1998) compare the underlying models and techniques in DSS to an earlier survey (Eom and Lee 1990) of DSS applications (1971-1988). Most notable observations include:

- Artificial intelligence tools and techniques are most frequently embedded in decision support systems.
- Multicriteria decision-making models are becoming the most widely embedded management science / operations research (MS/OR) tools in DSS.

Shim *et al.* (2002) confirm such observations. They indicate that the utilization of MCDM in DSS applications is consistent with Keen's (1987) "the next decade of DSS" second point on the application of analytic models and methods. Looking ahead to the year 2007, Shim *et al.* (2002) expect a continued emphasis on advanced mathematical programming software integrated with (for instance) MS Excel™. On the other hand, the use of Artificial Intelligence (Keen's third point) is being replaced by intelligent systems and soft computing. Moreover, in another study, Eom and Min (1999) examines the contribution of MCDM to DSS research. They conclude that MCDM researchers have made significant contributions to DSS subspecialties such as group support systems, model management, design and foundations, and multicriteria decision support systems (MCDSS).

In spite of the proliferation of artificial intelligence and MCDM in various DSS application domains, IDSS in aquaculture planning in nonexistent and MCDM-based DSS is limited. Specifically, El-Gayar and Leung (2000) describe the design and implementation of an MCDM-based DSS for regional aquaculture planning.

While allowing for the incorporation of multiple developmental criteria, the system is not capable of handling the inherent imprecision and uncertainty encountered in planning for regional aquaculture planning.

11.2.2 MCDM Models for Regional Aquaculture Development

Many MCDM models have been widely used in the planning, management, and evaluation of biocomplex systems. Examples from agriculture include models for operational as well as strategic planning purposes such as (Bazaraa and Bouzaher 1981, Romero and Rehman 1989, Tabucanon 1990, Tapia 1990, Qingzhen *et al.* 1991, Guo and He 1999). In water resources systems planning, MCDM models pursued objectives such as sustainable land development, water resources conversation, and water quality management (Chang *et al.* 1997). However, there are only two applications to aquaculture (Sylvia and Anderson 1993, El-Gayar and Leung 2001). Moreover, none of the aforementioned research addressed the uncertainty in the planning process.

While stochastic programming can be used to supplement conventional mathematical programming techniques in dealing with system uncertainties, the large quantities of data required for the identification of the underlying probability distributions render such techniques infeasible in many real-world situations. Moreover, not all sources of uncertainties are stochastic in nature, but are the result of vagueness associated with the measurement and interpretation of linguistic variables. Accordingly, fuzzy set theory and fuzzy MCDM have received wide attention in the planning, management and evaluation of biocomplex systems.

11.2.3 Fuzzy Logic in Sustainable Development

With respect to sustainable development, Cornelissen *et al.* (2000) illustrate the use of fuzzy set theory to assess sustainable development. Here an attempt is made to link human expectations about development, expressed in linguistic propositions, to numerical data, expressed in measurements of sustainability indicators through fuzzy set theory. Two models to assess the contribution of sustainability indicators (SI) to SD are presented. The first model applies a fuzzy set aggregation scheme, while the second model applies approximate reasoning. Both models are explored using a hypothetical example of housing systems for laying hens.

Moreover, Andriantiatsaholiniaina and Phillis (2000) advocate the use of fuzzy logic for the assessment of sustainability for selected economies at the national level. In their model, they evaluated overall sustainability based on eight secondary and two primary linguistic variables. The results indicate that no country exceeded 50% of overall development sustainability due mainly to a bad condition of its ecological system.

In agriculture, Dunn *et al.* (1995), and Marks *et al.* (1995) apply fuzzy sets to the problem of agricultural sustainability. They emphasize how fuzzy sets can mitigate some of the limitations inherent in MCDM techniques, such as the treatment of noncommensurate units, the ranking procedure for a solution, and the degree of discrimination of attribute values. A simple illustration is used to show how fuzzy systems can be used to compare the sustainability of two or more farming systems.

11.2.4 Fuzzy MCDM in Aquaculture and Related Areas

With respect to fuzzy optimization, Rommelfanger (1996) provides a representative list of fuzzy linear programs (FLP) that were developed to tackle problems encountered in real-world applications. In this list, only a handful deal with biocomplex systems, namely, Leung (1988) and Mjelde (1986) for regional resource allocation, Slowinski (1986) and (1987) for water-supply planning, Sommer and Pollatschek (1978) for air-pollution regulation, and Oder and Rentz (1993) for energy emission. Moreover, Fuller and Carlsson (1996) provide a review of recent development of fuzzy MCDM with particular emphasis on four application areas: evaluation of weapon systems, a project maturity evaluation system implemented at Mercedes-Benz, selection of technology transfer strategy in biotechnology, and aggregation of market research data. Chang *et al.* (1997) provide an expanded list of applications focusing on water-resource planning as well as present a fuzzy multi-objective programming model for the evaluation of sustainable management strategies of optimal land development in the reservoir watershed. Their model accounted for a number of environmental and economic goals and was applied to the Tweng-Wen reservoir watershed in Taiwan. Huang *et al.* (2002) present an interval-parameter fuzzy integer programming model for the planning of regional solid waste management systems under uncertainty. The model allows for more in-depth analysis of tradeoffs between environmental and economic objectives as well as system optimality and reliability. Borges and Antunes (2003) present an interactive approach to deal with fuzzy multiple objective programming problems. The aim is to model the uncertainty and imprecision associated with the input-output coefficients in an energy-economy planning model.

It is evident that while the applications of fuzzy MCDM are numerous and diverse, the application in aquaculture planning is non existent. Accordingly, this chapter reports the successful technology transfer of fuzzy MCDM to a new problem domain, namely aquaculture planning, and illustrates its application in the aforementioned area using a case study from Egypt.

11.3 The System

The proposed system is consistent with Simon's (1977) framework for decision making, in which the decision-making process is comprised of four phases (intelligence, design, choice, and implementation). Specifically, the system is designed to aid the decision maker (DM) with the design, and choice phases. In the design phase, the system is designed to allow the DM to formulate and customize the aquaculture planning problem to the region under consideration by selecting the planning objectives, resources and other constraints, as well as providing region specific data. The system also allows the user to select a principle of choice, *i. e.* optimizing vs. satisfying by incorporating a variety of modeling approaches, *e. g.*, compromise programming (optimizing), and weighted goal programming (WGP) (satisfying). The system can also develop a set of alternatives either through sensitivity analysis or, in the case of two objectives; the system can generate the entire Pareto efficient set. By predicting and measuring outcomes, the system allows the DM to analyze and understand the consequences of various decision scenarios. In the choice phase, the system is designed to generate alternative solutions (scenarios) and to allow the user to compare and contrast among such scenarios through sensitivity analysis (what-if analysis).

In addition to supporting the design and choice phases of the decision-making process, we identify several system-level design requirements including flexibility, user friendliness, and affordability. Flexibility refers to the ability of the DM to tailor the model to their specific region and problem, the ability to select from a variety of models, and the ability to accommodate various solvers. User friendliness refers to the ease of interaction with the system and the learning curve associated with using the system. Affordability refers to the total cost of ownership, *i. e.* acquisition and maintenance of the system.

Realizing the aforementioned features entails a development environment and a system architecture that facilitate the realization of such features. A development environment (DSS generator) that satisfies such requirements is Microsoft Excel. Specifically, MS Excel includes a powerful application development environment that allows for the development of customized user-friendly interfaces, interfaces seamlessly with database management systems (DBMS), includes a powerful programming language (Visual Basic for Applications – VBA), and includes its own solvers as well as its support for third-party solvers through add-in components. Moreover, as part of the Microsoft Office suite, MS Excel is widely available. In effect, MS Excel provides powerful and flexible support at an affordable cost.

The underlying architecture of the system is shown in Figure 11.1. The architecture is comprised of three components (Sprague and Carlson 1982), a user-interface, a database, and a modeling component. The user-interface allows the DM to access and manipulate the modeling and database components as well as analyze various decision scenarios (scenario management). Figure 11.2 depicts the overall structure of the interface that is implemented by customizing MS Excel's user interface.

The database component is comprised of a database management system (DBMS) and a database containing all relevant data. Through the user interface, the DBMS allows the DM to create, retrieve, update, and delete data pertaining to various decision scenarios. Such data include region-specific data required by the modeling component as well as result data obtained from running the models. The database design is normalized to eliminate possibilities for data anomalies as well as to facilitate the ease of future maintenance and upgrades to the database.

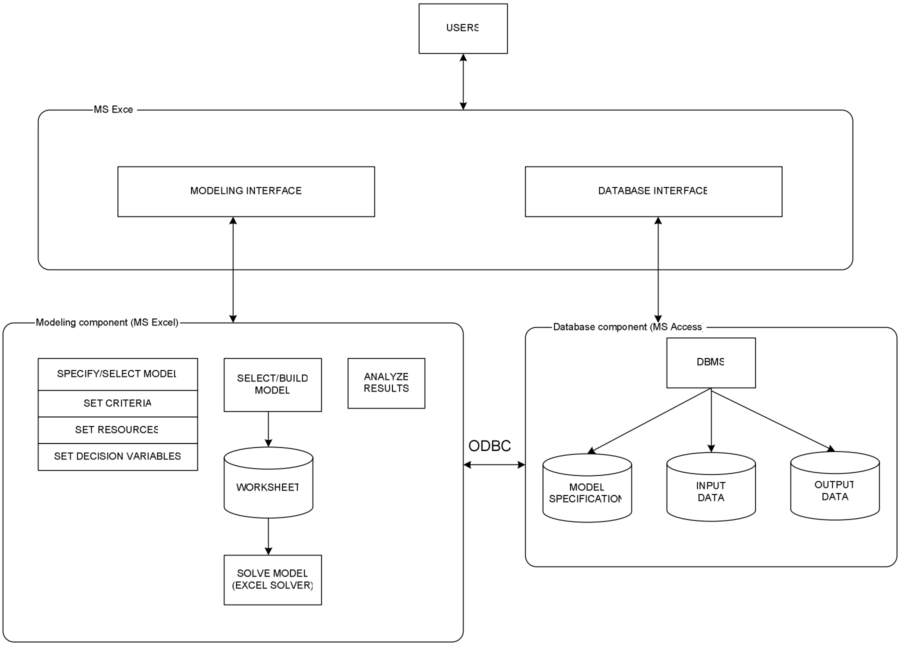


Figure 11.1. IDSS architecture

While MS Excel encompasses data-management capabilities, using a dedicated DBMS such as MS Access offers greater flexibility in database design and implementation. Moreover, MS Access is part of the MS Office suite and thereby integrates seamlessly with MS Excel. From an organizational perspective, extracting the data into an independent (standalone) database facilitates sharing the data with other applications. Figure 11.3 depicts the database interface.

Analogous to the database component, the modeling component allows the DM to create, retrieve, update, and delete models pertaining to the specific regional development problem under consideration. The modeling component is designed in a modular fashion with a core MCDM formulation defined in El-Gayar and Leung (2001) and depicted in Figure 11.4.

Several MCDM modeling paradigms can then be formulated utilizing the core. Examples of these formulations include multiple objective programming (MOP), weighted goal programming (WGP), and compromise programming (CP), and fuzzy multiple objective linear programming (FMOLP). Additional formulations may be added as the system evolves. Moreover, the modular design allows for model-data independence as well as model-solver independence. Both features are significant contributions to the system's flexibility. Within the implemented framework, the modeling component allows the DM to select criteria (*e. g.* Figure 11.5), resources, decision variable, and constraints relevant to the problem at hand. The component also allows the user to select a particular model formulation and provide data pertaining to that formulation, *e. g.* membership functions for FMOLP (Figure 11.6).

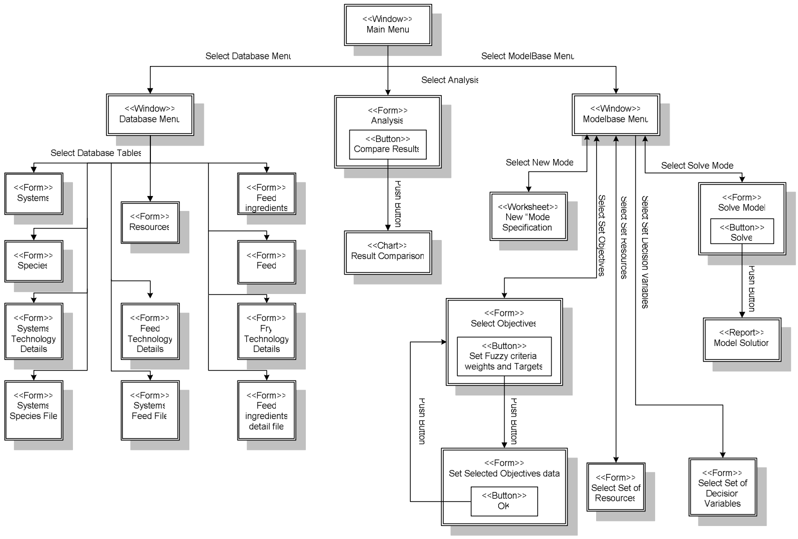


Figure 11.2. IDSS interface structure diagram

Figure 11.3. The database interface

The modeling component is implemented in MS Excel for ultimate flexibility and affordability. While the current version utilizes the Excel solver, the formulation does not preclude the utilization of dedicated solvers.

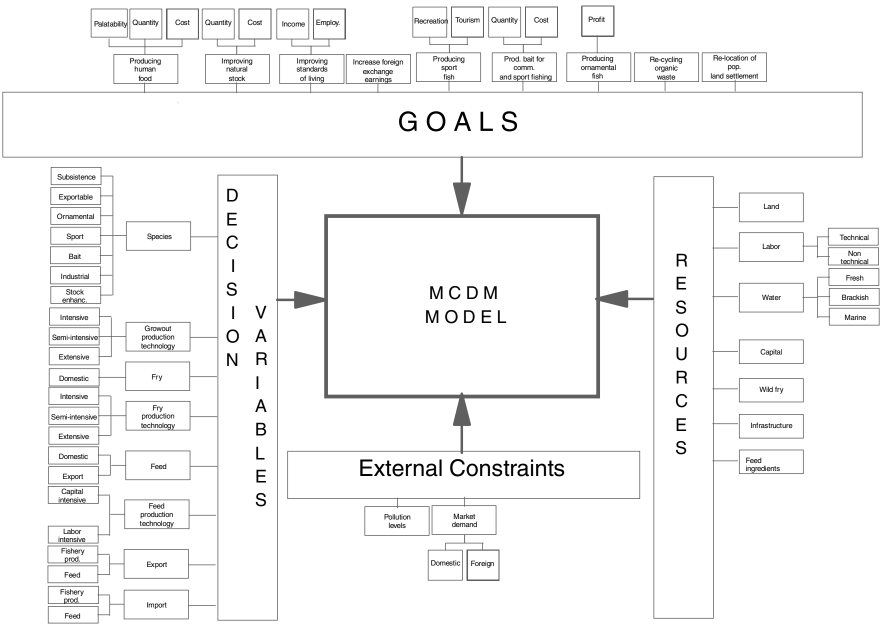


Figure 11.4. A MCDM framework for regional aquaculture development (El-Gayar and Leung 2001)

11.3.1 The Fuzzy MCDM model

The field of regional planning is characterized by the large number of issues and attributes involved, and regional planning for aquaculture development is no exception.

According to the MCDM framework (Figure 11.4) proposed by El-Gayar and Leung (2001), in aquaculture development, a planner or a policy maker is often confronted with a multitude of goals and objectives that he/she seeks to realize through the development of the region's aquaculture industry. Examples of such goals include producing human food, improving natural stock, improving the standards of living, and increasing foreign-exchange earnings. A development plan is thus comprised of the level of activities (decision variables) that would compromise among such often conflicting goals. Examples of such decision variables include what species to grow, what technology to use, how much to grow of each species and/or technology, *etc.* However, in devising such a plan, a set of resource, market and environmental constraints limit the alternatives available to the planner. Examples of such constraints include land, labor, water, *etc.*, as well as other external constraints such as domestic-market demand, export-market demand, and pollution restraints. The MCDM model thus seeks to assist the planner in identifying feasible (*i. e.* satisfying resources and external constraints) alternative development plans that attempt to reach a balance among the multiple goals and objectives encountered when planning for regional aquaculture development.

Figure 11.5. Modeling component - Selecting decision attributes

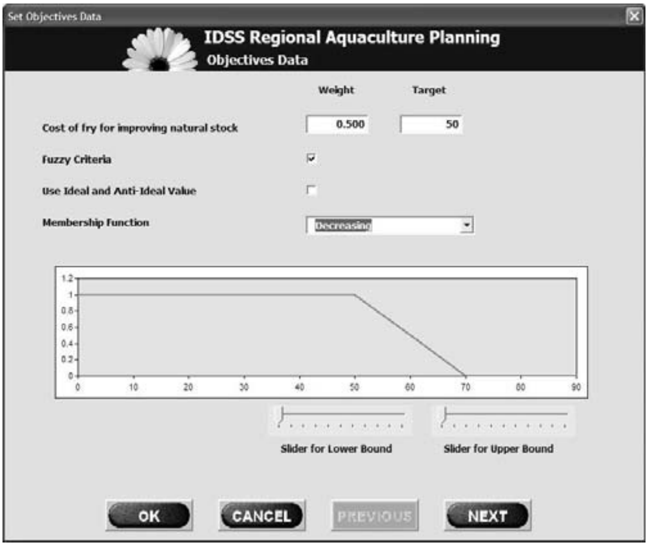


Figure 11.6. Defining membership functions for fuzzy multiobjective optimization

However, aquaculture plans do not have clearly defined objectives and fail to distinguish between objectives, strategies, and policies (Nash 1995). By extending El-Gayar and Leung's (2001) framework to incorporate imprecise/uncertain information, the proposed fuzzy MCDM modeling framework could certainly alleviate some of these issues as raised by Nash. In effect, in MCDM problems, a decision maker seeks to optimize multiple decision criteria simultaneously subject to a number of constraints. In general, and without loss of generality, we can describe an MCDM problem as:

$$\text{Max. } \mathbf{Z}(\boldsymbol{X}), \quad \text{Subject to,} \tag{11.1}$$
$$\mathbf{F}(\boldsymbol{X}) \leq \boldsymbol{B}, \text{ and } \mathbf{X} \geq 0$$

where, $\mathbf{Z}(\boldsymbol{X})$ is a $\boldsymbol{K}$ **x 1** vector of decision attributes, $\boldsymbol{K}$ is the number of attributes, $\mathbf{F}(\boldsymbol{X})$is a $\boldsymbol{M}$ **x 1** vector of constraints, $\boldsymbol{M}$ is the number of constraints, $\boldsymbol{B}$ is a $\boldsymbol{M}$ **x 1** vector of resource/capacity limits, and $\boldsymbol{X}$ is a $\boldsymbol{N}$ **x 1** vector of decision variables, $\boldsymbol{N}$ is the number of decision variables.

A linear MCDM model is then a model in which $\mathbf{Z}(\boldsymbol{X})$ and $\mathbf{F}(\boldsymbol{X})$ are linear. Such models are commonly referred to as multiobjective linear programming (MOLP) models and can be expressed as:

$$\text{Max. } \mathbf{Z}(\boldsymbol{X}) = \boldsymbol{C}\,\boldsymbol{X}, \quad \text{subject to,} \tag{11.2}$$
$$\mathbf{F}(\boldsymbol{X}) = \boldsymbol{A}\,\boldsymbol{X} \leq \boldsymbol{B} \text{ and } \boldsymbol{X} \geq 0$$

where, $\boldsymbol{C}$ is a $\boldsymbol{K}$ **x** $\boldsymbol{N}$ matrix of objective function coefficients, and $\boldsymbol{A}$ is a $\boldsymbol{M}$ **x** $\boldsymbol{N}$ matrix of technical coefficients.

Several techniques are available for handling MCDM problems (Zeleny 1982, Romero and Rehman 1989; Tabacanon 1988) to name a few. Such techniques vary in their suitability to handle different decision situations. However, in aquaculture planning, it is often the case where the available (and required) information contains uncertainty. Moreover, the information may not be sufficient to assess the probability distributions required for stochastic programming. In that regard, by describing the matrices $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ as fuzzy numbers, fuzzy logic allows us to explicitly model the uncertainty inherent in planning for regional aquaculture development. The resultant model is often referred to as a fuzzy MOLP (FMOLP) model and can be expressed as:

$$\mathbf{Z}(\boldsymbol{X}) = \boldsymbol{C}\,\boldsymbol{X} \geq \boldsymbol{G}, \quad \text{Subject to,} \tag{11.3}$$
$$\mathbf{F}(\boldsymbol{X}) = \boldsymbol{A}\,\boldsymbol{X} \leq \boldsymbol{B}, \text{ and } \boldsymbol{X} \geq 0$$

where, $\boldsymbol{G}$ is a $\boldsymbol{K}$ **x 1** vector of aspiration levels for each decision attributes, while $\geq$ and $\leq$ denote the fuzzified version of $\geq$ and $\leq$, respectively, and have the linguistic interpretation of "approximately larger than or equal to" and "approximately smaller than or equal to"

Since Bellman and Zadeh's (1970) seminal paper on decision making in a fuzzy environment, and Zimmerman's (1978) paper on fuzzy linear programming, several techniques have evolved to handle FMOLP (Luhandjula 1989, Rommelfanger 1996, Fuller and Carlsson 1996). Such techniques vary depending on the assumptions concerning the matrices $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ as well as the shapes of the underlying membership functions. In general, by defining membership functions $\boldsymbol{\mu}_{Gi}$ and $\boldsymbol{\mu}_{Cj}$ on both the $\boldsymbol{K}$ goals and the $\boldsymbol{M}$ constraints, respectively, a fuzzy decision $\boldsymbol{D}$ (viewed as a fuzzy set μ_D) can be defined as the confluence of goals and constraints. Such confluence can be represented as the intersection of all fuzzy sets as follows:

$$\begin{aligned}\mu_D &= \mu_{G_1} \wedge \ldots \wedge \mu_{G_i} \wedge \ldots \mu_{G_k} \wedge \mu_{C_1} \wedge \ldots \wedge \mu_{C_{1j}} \wedge \ldots \wedge \mu_{C_m} \\ &= Min(\mu_{G_1}, \ldots, \mu_{G_i}, \ldots, \mu_{G_k}, \mu_{C_1}, \ldots, \mu_{C_{1j}}, \ldots, \mu_{C_m})\end{aligned} \tag{11.4}$$

Notice that in this formulation, the conventional distinction between objectives and goals no longer exist. An optimal decision $\boldsymbol{D}^M$ is the nonfuzzy, but subset of $\boldsymbol{D}$ corresponding to an alternative in $\boldsymbol{X}$ that maximizes $\boldsymbol{\mu_D}$, *i. e.*,

$$Max(\mu_D) = Max(Min(\mu_{G_1}, \ldots, \mu_{G_i}, \ldots, \mu_{G_k}, \mu_{C_1}, \ldots, \mu_{C_{1j}}, \ldots, \mu_{C_m})) \tag{11.5}$$

In general, linear non-increasing and non-decreasing membership functions are frequently used for the inequalities "≤" and "≥", respectively (Chang *et al.* 1997). The main advantage of using linear membership functions is that it allows for the solution of the FMOLP model using conventional mathematical programming techniques. The fuzzy membership function corresponding to "≥" that is used in this model is shown in Figure 11.7 and can be expressed as:

$$\mu_{G_i} = \begin{cases} 0 & \text{if } Z_i \le Z_i^w \\ \dfrac{Z_i - Z_i^w}{\Delta_{G_i}} & \text{if } Z_i^w < Z_i < Z_i^b \\ 1 & \text{if } Z_i \ge Z_i^b \end{cases} \tag{11.6}$$

where, Z_i^b and Z_i^w are the aspiration level (goal) and anti-ideal (worst value) for the i^{th} decision attribute, respectively. If the decision maker seeks the best possible level, the goal corresponds to the ideal value (best value) for the i^{th} objective, while Δ_{G_i} is equal to $Z_i^b - Z_i^w$ and reflects the tolerance of the decision maker.

Note that both the ideal and anti-ideal values can be determined from the payoff matrix obtained by solving $\boldsymbol{K}$ deterministic single objective linear programming

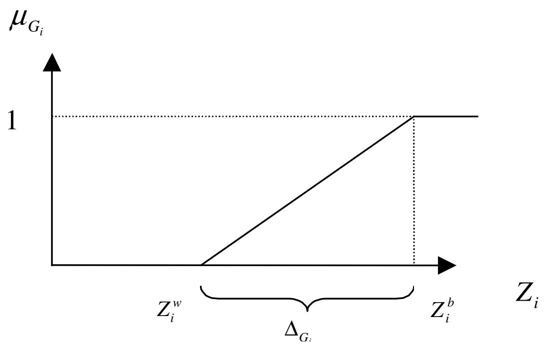


Figure 11.7. Fuzzy nondecreasing membership function corresponding to ≥

problems. The fuzzy membership function corresponding to "≤" that is used in this model is shown in Figure 11.8 can be expressed as:

$$\mu_{C_j} = \begin{cases} 1 & \text{if } F_j \le F_j^b \\ 1 - \dfrac{F_j - F_j^b}{\Delta_{C_j}} & \text{if } F_j^b < F_j < F_j^w \\ 0 & \text{if } F_j \ge F_j^w \end{cases} \tag{11.7}$$

Where, F_j^b and F_j^w are the most and least preferred values for the j^{th} constraint, respectively, while Δ_{C_j} is equal to $F_j^w - F_j^b$.

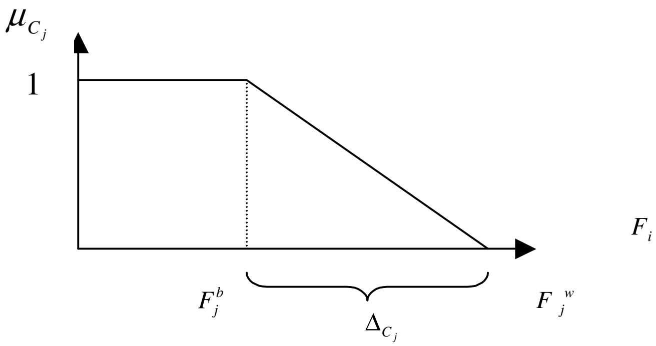


Figure 11.8. Fuzzy nonincreasing membership function corresponding to ≤

Using (11.6) and (11.7), and substituting in (11.4) and (11.5), the problem is equivalent to solving the following crisp linear programming problem:

$$Max \quad \lambda, \quad \text{subject to,}$$

$$\frac{Z_i - Z_i^w}{\Delta_{G_i}} \ge \lambda \qquad \text{i} = 1,2,\ldots\text{k} \tag{11.8}$$

$$1 - \frac{F_j - F_j^b}{\Delta_{C_j}} \ge \lambda \quad \text{j} = 1,2,\ldots\text{m, and } X \ge 0$$

To accommodate the preferences of the decision maker, the model can be reformulated as (Martinson 1993):

$$Max \quad \lambda, \text{ subject to,}$$

$$\frac{1}{W_{G_i}} \cdot \left[\frac{Z_i - Z_i^w}{\Delta_{G_i}} \right] \geq \lambda \qquad i = 1,2,...k \tag{11.9}$$

$$\frac{1}{W_{C_j}} \cdot \left[1 - \frac{F_j - F_j^w}{\Delta_{C_j}} \right] \geq \lambda \qquad j = 1,2,...m, \text{ and } X \geq 0$$

where, W_{Gi} and W_{Cj} reflect the relative importance of the i^{th} decision attribute and the j^{th} constraint, respectively.

11.4 Case Study

11.4.1 Background

The case study is concerned with resolving issues pertaining to the planning for aquaculture development in Northern Egypt where aquaculture is considered as a viable industry with significant potential for supplying cheap and good-quality protein, for helping to balance the foreign-exchange deficit, and for creating employment opportunities. However, Northern-Egypt is characterized by being able to accommodate a wide variety of aquaculture production systems and technologies. Such production systems incorporate a variety of species, primarily sea bream, sea bass, mullet, tilapia and carp that are grown separately (monoculture) or in combinations (polyculture). Moreover, production systems can be employed at different technology levels ranging from intensive to semi-intensive to extensive culture systems.

The problem is further complicated as systems and technologies vary in yield, product quality, profit and utilization of resources. Accordingly, the choice of systems and the level of technology to employ would clearly affect the levels of the different decision attributes with the possible need to compromise between these attributes. Moreover, similar issues pertain to the choice of fry and feed production systems and technologies. Export and import decisions such as identifying how much to import and how much to export from each species and feed type also play a role.

In formulating the decision problem, the DM finds it difficult, if not impossible, to provide target values for decision attributes with precision. Moreover, the DM preference does not necessarily reflect a clear cutoff value for a particular target.

Similar issues arise with resource constraints as well. This is where fuzzy logic is leveraged to address such issues.

11.4.2 Model Formulation

For the particular case study under consideration, three decision attributes are of concern, namely, regional availability of protein, employment, and foreign-exchange earnings. The resources included are: land for fresh, brackish, and marine aquaculture, technical and nontechnical labor, fresh, brackish, and marine water, domestic capital, foreign reserves, fry constraints, feed constraints, and feed ingredient constraints. The external constraints represent domestic-market demand, export-market demand, and pollution. The resource constraints together with the external constraints define the feasible set ***F*** of solutions.

11.4.3 Using the System

Using the database component (Figure 11.3), the user is able to enter region specific data pertaining to aquaculture planning. Examples include data pertaining to the cultured species, the available aquaculture technologies as well as technologies for producing feed ingredients, and market data such as demand and prices by species.

For the specific planning scenario under consideration, and using a screen similar to Figure 11.5, the user selects the specific objectives (three in this case study), the decision variable, and the constraints to include in the scenario. The user can manipulate the shape of the membership functions using screens similar to Figure 11.6.

11.4.4 Results and Discussion

This case study is comprised of three scenarios. The scenarios differ with respect to the parameter values for the fuzzy membership functions as well as the weight assigned to the different objectives under consideration. In the first scenario, and for the purpose of practical implementation, we use the results of the payoff matrix to determine the tolerance intervals for the fuzzy membership functions. Table 11.1 shows the payoff matrix for the three objectives and the ideal (best) and anti-ideal (worst) values for each of the objectives. Each column of the payoff matrix corresponds to a single objective and represents the values for the three objectives when that objective is optimized separately (using the deterministic MCDM model).

The diagonal of the payoff matrix thus represents the ideal vector. Table 11.2 shows the results for the first scenario. Overall, the solution strikes a balance among the competing objectives in which regional availability of protein, employment, foreign exchange earning attains 61%, 86%, and 54% of their ideal value, respectively.

In the second scenario, and to test the flexibility in decision making, different weight distributions are applied to the objectives under consideration. Specifically, in this scenario foreign-exchange earnings is four times as important as the other two objectives. As shown in Table 11.3, the solution reflects that change in preference in which foreign-exchange earnings attain 93% of its ideal value at the expense of the other two objectives.

Table 11.1. The payoff matrix and the ideal and anti-ideal values

The objectives	Objective to optimize			Ideal	Anti-ideal
	1	2	3		
Regional availability of protein (Kton)	88	83	6	88	6
Employment (jobs)	3527	5088	1159	5088	1159
Foreign-exchange earnings (M LE)	-10	-2	104	104	-10

Table 11.2. The solution for scenario 1

Objectives	Weight	FMOLP	Ideal	Anti-ideal
Regional availability of fish in thousand tons	1	54	88	6
Employment in jobs/year	1	4349	5088	1159
Foreign-exchange earnings in million LE	1	57	104	-10

Table 11.3. The solution for scenario 2

Objectives	Weight	FMOLP	Ideal	Anti-ideal
Regional availability of fish in thousand tons	1	25	88	6
Employment in jobs/year	1	2304	5088	1159
Foreign-exchange earnings in million LE	4	97	104	-10

It should be noted, however, that the ideal and anti-ideal numbers represent extreme cases, and may not necessarily reflect the preference of the decision maker. To illustrate the flexibility of FMOLP in accommodating uncertainty, the third scenario represents a case where the tolerance interval is decreased by 25 percent relative to the first scenario. The relative importance of the objectives remains unchanged from the second scenario. By decreasing the tolerance interval, we effectively increase the slope of the membership function and increase the anti-ideal (lower limit of what is acceptable to the decision maker). As shown in Table 11.4, the solution reflects the decision-maker preferences by increasing the level of attainment of regional availability of protein at the expense of foreign-exchange earning.

Table 11.4.. The solution for scenario 3

Objectives	Weight	FMOLP	Ideal	Anti-ideal
Regional availability of fish in thousand tons	1	38	88	27
Employment in jobs/year	1	2998	5088	2141
Foreign-exchange earnings in million LE	4	80	104	19

11.5 Conclusions

This chapter presents the development of an IDSS that applies fuzzy set theory to multiple criteria decision making in aquaculture planning. In effect, the chapter demonstrates how fuzzy set theory can be used to explicitly account for the inherent uncertainty encountered when planning for aquaculture development in a given region. The case study illus ates the use of the proposed IDSS and fuzzy MCDM framework to aquaculture planning in Egypt.

Overall, the proposed IDSS and fuzzy MCDM framework constitutes an intellectually appealing analytical framework as it simultaneously accommodates uncertainty and multiplicity of developmental objectives in the context of regional aquaculture planning. Moreover, the user-friendly decision support environment allows the decision maker to interactively change various model parameters, including fuzzy membership function types and parameters. For fuzzy and MCDM theories, biocomplex systems in general and aquaculture systems in particular offer challenging application domains for testing and further development of such theories.

11.5.1 Recommendation for Future Research

The proposed research can be extended in various way. Examples include:

- The derivation of fuzzy membership functions that closely represent the decision-maker's preferences. Such functions are not necessarily linear as represented in the proposed framework. Whether the improvement in the results warrants the additional complexity from utilizing nonlinear fuzzy membership function requires further investigation.
- Capitalizing on IDSS's architecture and extending the model base by adopting existing as well as developing new interactive fuzzy MCDM techniques appropriate for planning for regional aquaculture development.

Acknowledgements

This material is based upon work supported by the National Science Foundation/EPSCoR Grant #EPS-0091948 and by the State of South Dakota. The usual disclaimer applies.

References

Andriantiatsaholiniana LA, Phillis YA (2000) S.A.F.E Sustainability assessment by fuzzy evaluation. Retrieved April 2003 from http://www.soc.uoc.gr/calendar/2000EAERE/papers/PDF/G6-Andriantiatsaholiniaina.pdf

Bazaraa M, Bouzaher A (1981) A linear goal programming model for developing economies with an illustration from the Agricultural sector in Egypt, Management Science 27: 396-413

Bellman RE, Zadeh LA (1970) Decision-making in a fuzzy environment. Management Science 17: 141-164

Borges AR, Antunes CH (20 A fuzzy multiple objective decision support model for energy-economy planning. European Journal of Operational Research 14: 304-316

Carlsson C (1999) Soft computing: modeling technologies and intelligent systems Minitrack. Proceedings of the 32nd Hawaii International Conference on System Sciences, Honolulu, Hawaii, January 5-8.

Chang N, Wen CG, Chen YL (1997) A fuzzy multi-objective programming approach for optimal management of the reservoir watershed. European Journal of Operational Research 99: 289-302

Cornelissen AMG, Van Den Berg J, Koops WJ, Grossman M, Udo HMJ (2000) Assessment of sustainable development: a novel approach using fuzzy set theory. Erasmus Research Institute of Management (ERIM), report number ERS-2000-23-LIS

Cox ED (1995) Fuzzy Logic for Business and Industry. Charles River Media, Inc., Hingham, MA

Dunn EG, Keller JK, Marks LA, Ikerd JE, Gader PD, Godsey LD (1995) Extending the application of fuzzy sets to the problem of agricultural sustainability. Proceedings of the International Symposium of Uncertainty Modeling and Analysis (ISUMA-NAFIPS '95). College Park, MY

El-Gayar OF, Leung PS (2000) ADDSS: a tool for regional aquaculture development. Aquacultural Engineering 23: 181-202

El-Gayar OF, Leung PS (2001) A multiple criteria decision-making framework for regional aquaculture development. European Journal of Operational Research 133: 462-482

Eom HB, Lee SM (1990) A survey of decision support system applications (1971-1988), Interface, 20(3): 65-79.

Eom SB, Min H (1999) The contributions of multi-criteria decision-making to the development of decision support systems subspecialties: an empirical investigation. Journal of Multicriteria Decision Analysis 8: 239-255

Eom SB, Lee SM, Kim EB, Somarajan C (1998) A survey of decision support system applications (1988-1994). Journal of the Operational Research Society 49: 109-120

Food and Agriculture Organization (FAO) of the United Nations (2003) Farming fish for the future, sustainability FAO Sub-Committee on Aquaculture, Norway, Rome http://www.fao.org/english/newsroom/news/2003/21619-en.html

Fuller R, Carlsson C (1996) Fuzzy multiple criteria decision-making: Recent developments. Fuzzy Sets and Systems 78: 139-153

Guo L, He Y (1999) Integrated multi-criteria decision model: a case study for the allocation of facilities in Chinese agriculture. Journal of Agricultural Engineering Research 73 : 87-95

Huang YF, Baetz BW, Huang GH, Liu L (2002) Violation analysis for solid waste management systems: an interval fuzzy programming approach. Journal of Environmental Management 65: 431-446

Keen P (1987) Decision support systems: the next decade. Decision Support Systems 3:253-265

Leung Y (1988) Interregional equilibrium and fuzzy linear programming. Environment and Planning A 20: 25-40

Luhandjula MK (1989) Fuzzy optimization: an appraisal. Fuzzy sets and systems 30: 257-282

Marks LA, Dunn EG, Keller JK, Godsey LD (1995) Multiple criteria decision-making (MCDM) using fuzzy logic: An innovative approach to sustainable aquaculture. Proceedings of the International Symposium of Uncertainty Modeling and Analysis (ISUMA-NAFIPS '95). College Park, MY

Martinson FK (1993) Fuzzy v.s. minmax weighted multiobjective linear programming illustrative comparison. Decision Science, 24(4): 809-824

Mjelde KM (1986) Fuzzy resource allocation. Fuzzy Sets and Systems 19: 239-250

Nash CE (1995) Aquaculture Sector Planning and Management. Fishing News Books, Oxford, England

Oder C, Rentz O (1993) Entwicklung eines auf der Theorie unscharfer Mengen basierenden Energie-Emissions-Modells. In: Operations Research Proceedings 1992, Springer-Verlag Berlin, pp 111-118

Qingzhen Z, Changyu W, Zhimin Z (1991) The application of operations research in optimization of agricultural production. Operations Research 39 : 194-205

Romero C, Rehman T (1989) Multiple Criteria Analysis for Agricultural Decisions. Elsevier Science Publishers, Amsterdam

Rommelfanger H (1996) Fuzzy linear programming and applications. European Journal of Operational Research 92: 512-527

Shim JP, Warkentin M, Courtney JF, Power DJ, Sharda R, Carlsson C (2002) Past, Present, and future of decision support technology. Decision Support Systems 33: 111-126

Simon H (1977) The New Science of Management Decision. Englewood Cliffs, NJ

Slowinski R (1986) A multicriteria fuzzy linear programming method for water supply system development planning. Fuzzy Sets and Systems 19: 217-237

Slowinski R (1987) An interactive method for multiobjective linear programming with fuzzy parameters and its application to water supply planning. In: Kacprzyk J and Orlovski SA (Eds.) Optimization Models Using Fuzzy Sets and Possibility Theory, Reidel Dordrecht, pp 396-414

Sommer G, Pollatschek MA (1978) A fuzzy programming approach to an air pollution regulation problem. In: Trappl R, Klir GJ , Pichler FR (Eds.) Progress in Cybernetics and Systems Research: General System Methodology, Mathematical System Theory, and Fuzzy Systems, Biocybernetics and Theoretical Neurobiology, Hemisphere Washington, DC, pp 303-313

Sprague RH Jr, Carlson ED (1982) Building Effective Decision Support Systems Englewood Cliffs, NJ

Sylvia G, Anderson JL (1993) An economic policy model for net-pen salmon farming. In: Hatch U and Kinnucan H (Eds.) Aquaculture: Models and Economics, Westview Press, Boulder, pp. 17-38

Tabucanon MT (1990) Multiple criteria decision-making framework for food policy analysis, In: Proceedings of the International Conference on Operations Research/Management Science: Techniques and Applications, Manila, Philippines

Tabucanon MT (1988) Multiple Criteria Decision-making in Industry. Elsevier Science Publishers, Amsterdam

Tapia CG (1990). A multiobjective agricultural planning model, in Proceedings of the International Conference on Operations Research/Management Science: Techniques and Applications, Manila, Philippines

World Commission on Environment and Development (1987) Our Common Future (The Brundtland Report). Oxford University Press, Oxford

Zeleny M (1982).Multiple Criteria Decision-making. McGraw Hill, New York

Zimmerman HJ (1978) Fuzzy programming and linear programming with several objective functions. Fuzzy Sets and Systems 1: 45-55

12

An i-DMSS Based on Bipartite Matching and Heuristics for Rental Bus Allocation

Katta G. Murty[1] and Woo-Je Kim[2]

[1] Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, Michigan 48109-2117, USA, e-mail:murty@umich.edu
[2] Department of Industrial and Information Systems Engineering, Seoul National University of Technology, 172 Gongneung 2-dong, Nowon-gu, Seoul 139-743, South Korea, e-mail:wjkim@snut.ac.kr

Faced with the imminent retirement of two senior employees who used to make decisions on bus allocations to customers manually, a bus rental company in Seoul, South Korea, asked us to develop a DMSS (decision-making support system) to help the young fresh graduate employee who will be taking over this job from them. Practice has shown that allocation and routing decisions made manually by human operators with long experience are usually nearly optimal, and it is very hard to beat those decisions using a computerized DMSS. Therefore the company asked us to design an i-DMSS (intelligent DMSS) that can help the new decision maker to reach decisions comparable in quality to those made by the retiring pair of senior decision makers. In this paper we discuss this decision problem, its context, the models we used to solve it, the algorithms we used in the i-DMSS to solve these models, and how this i-DMSS is used to make the decisions daily. The i-DMSS is based on bipartite matching and transportation algorithms and heuristics, and produces solutions 10-20% more economical than the manual decisions.

12.1 Introduction: Bridging the Gap Between Theory and Practice

ORMS theory has developed efficient algorithms for solving some single objective optimization models that are highly structured.

When practitioners try to apply these ORMS tools to solve real-world decision-making problems, they usually find that none of them applies to their problem exactly. Most real-world decision-making problems involve multiple objectives that need to be optimized simultaneously, also, often they lack the nice structure of the models studied in ORMS theory. As Wolfram suggests (2002) *"...the idea of describing behavior in terms of mathematical equations works well where the behavior is fairly simple. It almost inevitably fails whenever the behavior is more complex"*.

Hence there is a wide gulf between real-world problems, and mathematical models for which efficient algorithms have been developed in theory. To bridge this gulf and get good results, it is essential to model real-world problems intelligently. Heuristic modeling techniques, approximations, relaxations, hierarchical modeling techniques with substitute objective functions for each stage (Murty and Djang (1999), and heuristic algorithms serve as a bridge between the two sides of this gulf (Figure 12.1) (Murty (2005). This theoretical justification is also supported by Ackoff (1996), Geoffrion (1976), and Simon (1987).

In this mode of application, the methods developed in ORMS theory are not the main methods for solving complex real world problems, but become valuable tools for designing intelligent approaches to handle them.

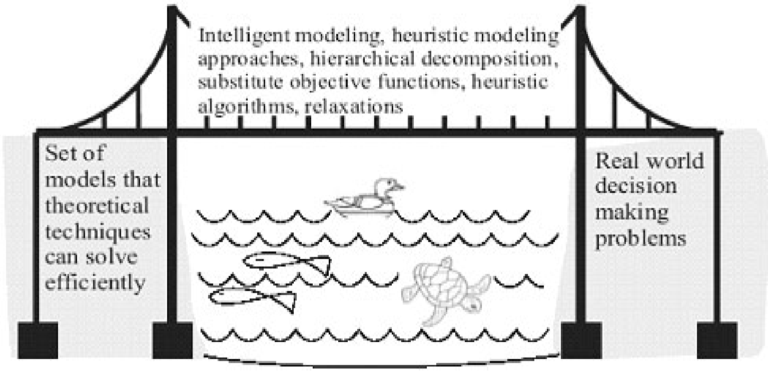


Figure 12.1. The wide gap between efficiently solvable mathematical models and real-world decision-making problems, and its bridge

We illustrate this process with a project to develop an i-DMSS (intelligent decisions-making support system) for rental bus allocation carried out for a small company in Seoul that rents buses with drivers to customer groups who request them.

12.2 The Problem and its Context

12.2.1 The Problem Description

We discuss the process of analyzing, modeling, and developing an i-DMSS to solve the bus-allocation problem. The application involves solving the same type of problem daily with new data for each day. This is a typical problem that arises at bus- rental companies in major cities all over the world, even though the constraints and other features of the problem may vary from company to company.

This company rents buses (with drivers provided) to customer groups who request them. The group size varies considerably. To serve customer needs economically, the company rents two size buses; a small 15-seat bus for small groups (they have five of these buses), and a larger 45-seat bus for larger groups (they have 20 of these buses).

Each customer request (called **a job**) completely specifies the route that the group wants to take, the starting location and starting time, any intermediate stops in the job, and the ending location and the ending time. Since the customer completely specifies the route in their job, there is no routing to be done by the company.

We denote:

n = number of jobs for which buses are to be allotted on a day
t_{-i}, t^{-}_{i} = starting and ending clock times of job i, $i = 1$ to n
p_i, q_i = starting and ending locations of job i, $i = 1$ to n
L = set consisting of the two bus depots, and all the distinct sites among the starting and ending locations of the n jobs
d_{ij} = distance from i to j measured in expected travel time in minutes, for i,j elements of L, $i \neq j$

The difference between the job ending and starting times, $t^{-}_{i} - t_{-i}$, is called the duration of job i. This varies between 0.5 h to 20 h, but more than 75% of the jobs this company receives have durations ≤ 5 h.

The $|L| \times |L|$ matrix $D = (d_{ij})$ is the distance data matrix for the problem for that day.

The company gets up to 100 requests each day. Jobs are classified into **large group jobs** and **small group jobs** depending on the corresponding customer group size. Large group jobs need a 45-seat bus, while a small group job can be serviced by either a 15-seat bus or a 45-seat bus. While servicing a job, the bus and its driver should be at the disposal of the customer corresponding to that job, *i. e.* two jobs cannot be combined into a bus at the same time. The company's charges for each customer depend on the size of the bus they need, the duration of their whole job, and the total mileage on that job.

The company keeps its buses at two depots in different locations. At each depot they have a staff of drivers for the jobs served by buses from that depot. On days when their own buses are not adequate, the company itself rents buses from other vendors. All the data about jobs to be served on a day is available at the company by the day before. They finalize all the bus and driver allocations for each day by the evening of the day before; so that the drivers can take the buses from their depots to the starting locations of the first jobs, on time. The drivers bring the buses back to their depots from the ending locations of their last jobs.

Since several jobs are of short duration, for economical operation the company likes to pack as many jobs as possible, one after the other, into each bus's daily work schedule known as its **work-sequence** for the day. Suppose a bus handles jobs numbered $i_1, \ldots, i_h$ in that order on a day, then its work-sequence for that day is the sequence of jobs ($\boldsymbol{i_1, \ldots, i_h}$). Then for each $\boldsymbol{g = 1}$ to $\boldsymbol{h - 1}$ after completing job $\boldsymbol{i_g}$ at clock time $\bar{t}_g$ at location $\boldsymbol{q_g}$, the driver has to drive that bus to the starting location $\boldsymbol{p_{g+1}}$ of the next job $\boldsymbol{i_{g+1}}$ before its starting time $\underline{t}_{g+1}$. So, for ($\boldsymbol{i_1, \ldots, i_h}$) to be a work-sequence, the condition: $\underline{t}_{g+1} - \bar{t}_g \geq$ driving time from $\boldsymbol{q_g}$ to $\boldsymbol{p_{g+1}}$ must hold for all $\boldsymbol{g = 1}$ to $\boldsymbol{h - 1}$. In this case, the drives from $\boldsymbol{q_g}$ to $\boldsymbol{p_{g+1}}$ for $\boldsymbol{g = 1}$ to $\boldsymbol{h - 1}$ of this bus are called **empty-load drives** on this work-sequence. During an empty-load drive the company is incurring the cost of keeping the bus running (fuel *etc.* + driver's wages) on its own without any customer paying for it.

All the buses start at their depot, and return to their depot after their last job in their work-sequence for the day. So for this bus the quantity [$\bar{t}_h - \underline{t}_1$ + (driving time from depot to $\boldsymbol{p_1}$) + (driving time from $\boldsymbol{q_h}$ to depot)] measured in hours represents the time in hours the driver of that bus worked that day, and this quantity is called the **duration of the work-sequence** ($\boldsymbol{i_1, \ldots, i_h}$), (note that this depends on the depot of the bus to which this work-sequence is assigned).

The driver's wages for a day are proportional to the number of hours s/he has worked. For this reason drivers have a strong desire to maximize the number of hours that they work, but driving fatigue can lead to serious accidents, that's why the company likes to keep the duration of work-sequences to less than 12 hours. Work-sequences of duration over 12 h are called **long-duration work-sequences**. However, there are some single jobs that are themselves of duration over 24 h, and these long-duration jobs are quite lucrative to the company. However, such jobs usually have many intermediate stops of considerable length during which the driver has nothing to do but wait, so s/he can rest, take a nap, and thus refresh her/himself. Long duration jobs and work-sequences with such intermediate rest periods do not contribute to fatigue, and hence should be considered differently from other long duration work-sequences involving fatigue-causing continuous driving. So, the company has decided to allow such long-duration work-sequences, but set it as a goal to keep the percentage of these long-duration work-sequences to 50 as far as possible so that they can alternate a long-duration work-sequence allotted to a driver one day with a short-duration work-sequence the next day.

The problem is a multiobjective problem. The most important objective is to minimize **OBJ1** = the total number of buses used to handle the jobs = the total number of work-sequences into which the jobs are partitioned. This also involves minimizing the expenses on renting other vendors buses used to handle the jobs. Customers pay for all the travel within the jobs, but the company does not collect any money for the travel from the depot to the starting location of the first job in the work-sequence, and back from the ending location of the last job in the work-sequence to the depot; and the travel from the ending location of a job to the starting location of the next job within a work-sequence. The second most important objective to minimize **OBJ2** = the total cost of empty-load driving of all the buses. The third most important objective stated as a goal is to keep **OBJ3** = the percentage of long-duration work-sequences, below 50 as far as possible.

12.2.2 How Were the Allocations Made in the Past?

The company had two full-time employees who were doing these allocations over a long period of time manually using a map of the various locations involved each day. This was their full time job at the company. With the imminent retirement of these experienced decision makers the company asked us to develop a computerized DMSS to help the new young person who will replace these two old timers.

12.3 Overall Approach for the i-DMSS

We found it very difficult to construct a single mathematical model encompassing all the features in the problem. Even if there is such a model, it will be very difficult to solve. To see this, consider the simpler problem of minimizing OBJ1 subject to the upper-bound constraint on the driving time of each driver in the case where the set of jobs to be handled contains no long-duration trips. This problem is equivalent to that of arranging all the jobs into the smallest number of **feasible work-sequences** where a feasible work-sequence is one satisfying the upper limit of 12 h on its duration. We show in Section 12.5.1 that this problem itself is NP-hard. So, we applied hierarchical decomposition to break up the problem into several stages that are simpler to model.

We first developed a procedure (**Procedure 1**) for solving the bus allocation problem ignoring the sizes of customer groups and buses, *i. e.*, considering only one size bus that fits all the groups. Here is a brief description of Procedure 1.

Procedure 1: Procedure when there is only one bus type

Stage 1: Partition the set of jobs into work-sequences (Section 12.5). This stage involves the following steps.
Step 1.1: Construct the job precedence acyclic network $\boldsymbol{G}$ and the bipartite network $\boldsymbol{B}$ corresponding to it (Section 12.5.1). Go to Step 1.2.
Step 1.2: Sep up the cost vector ($\boldsymbol{c_{ij}}$) corresponding to **OBJ2.1** (defined in Sections 12.5.2, 12.5.3) for edges in $\boldsymbol{B}$, and solve the min cost maximum cardinality matching problem in it using the min cost max flow algorithm, and from that matching find the initial set of work-sequences minimizing **OBJ1** and **OBJ2.1** (Section 12.5.2, 12.5.3).
Step 1.3: Revise the initial set of work-sequences using the heuristic approach (Section 12.5.4) if necessary to meet **OBJ3**. Go to Stage 2 with the final set of work-sequences.

Stage 2: Allocate buses to work-sequences (Section 12.6). This involves the following step.

Step 2.1: Set up the transportation model (Section 12.6) for allocating buses from depots 1,2 or outside vendors to the work-sequences, and solve it using the min cost flow algorithm. Terminate.

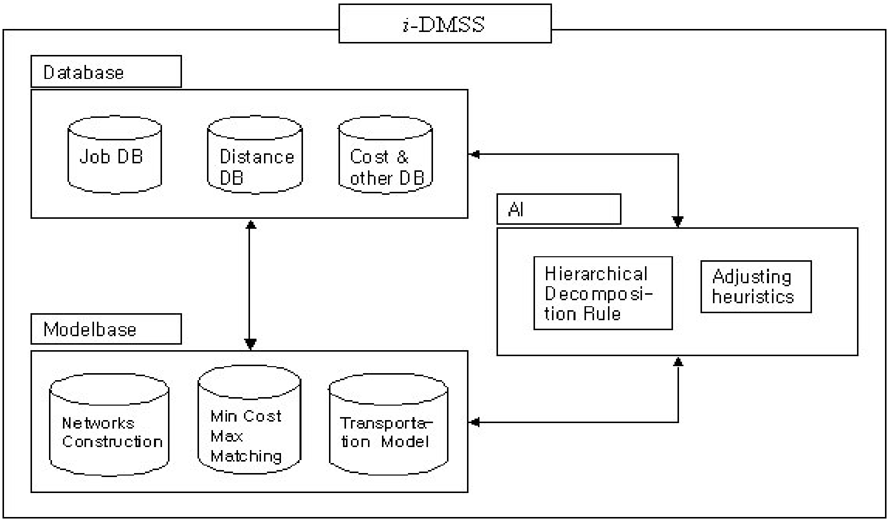


Figure 12.2. The structure of the i-DMSS

The i-DMSS consists of database(DB), modelbase, and artificial intelligence(AI) modules as in Figure 12.2. Database contains job DB, distance DB, cost and other DB, *etc.* Modelbase has the models that are used in the various steps, the networks construction model, the min cost max matching model to obtain the work-sequences, and the Transportation Model to allocate a bus to each work-sequence, and programs to solve these models. AI modules contain hierarchical decomposition rule and adjusting heuristics. Hierarchical decomposition rule has the rules for how to decompose the problem hierarchically. Adjusting heuristics have some modification rules for satisfying OBJ 3 and selection rules for choosing top the 5 work-sequences for small group jobs, *etc.*

In the next section we describe how the job classes and bus types are handled using this **Procedure 1** in two phases.

12.4 Decompositon for Two Types of Buses

In this problem, we have large group jobs and small group jobs, 15-seat buses that can serve small group jobs only, and 45-seat buses that can serve all the jobs. On some days when there are a large number of small group jobs, the company may experience a shortage of 15-seat buses to handle all of them. On such days, instead of renting some extra 15-seat buses from outside vendors, the company has found it to be much more economical to assign some of its own 45-seat buses to small group jobs. Because of this, we use the following hierarchical procedure for handling the allocation of two types of buses to the various jobs:

Phase 1: First consider only the small group jobs for which the 15-seat bus is suitable. Apply Procedure 1, Stage 1, to partition this subset of jobs into work-sequences. Find the total working time associated with each of the work-sequences. Select a threshold value, say **δ** h (currently **δ** = 6), as a lower bound for a day's working time. For all work-sequences associated with a working time ≥ **δ**, assign 15-seat buses to the extent they are available, using the model to be discussed in Section 12.6 with the only sources as the two depots of the company.

Phase 2: The jobs on all the work-sequences associated with working time < **δ**, and all the other work-sequences for which 15-seat buses may not have been allocated in Phase 1, are combined with the set of large group jobs. We then apply Procedure 1 to allocate 45-seat buses for this set of jobs.

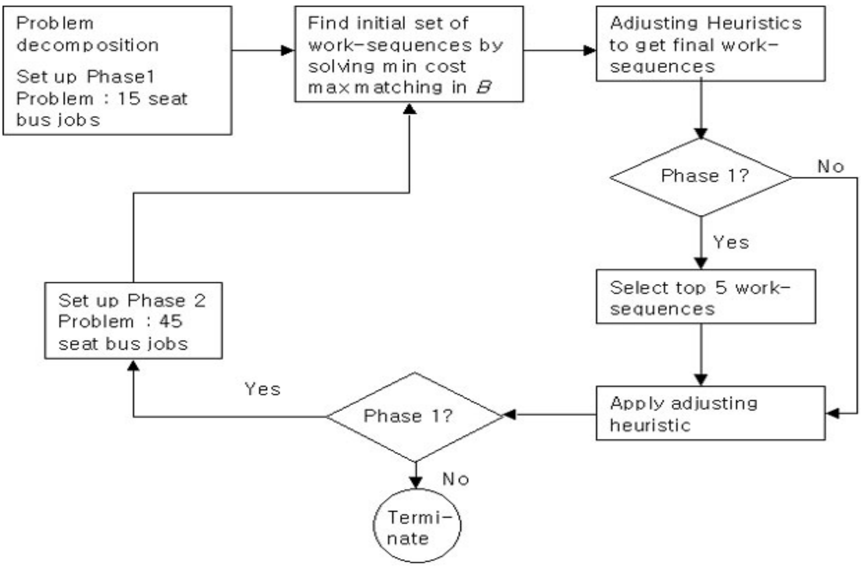


Figure 12.3. Overrall flowchart of the i-DMSS

12.5 The Model to Partition the Set of Jobs Into Work-sequences

12.5.1 How to Minimize OBJ1?

Minimizing **OBJ1** requires that we partition the set of n jobs into the smallest possible number of work-sequences. There is an efficient network model for this problem. For $i = 1$ to n, represent job i by node i in a directed network.

Job i is called a long-duration job if its duration $t'_i - t_i \geq 12$ h, nodes corresponding to such jobs are left as isolated nodes in the network without any arcs incident on them. These jobs are already too long, and we will not consider combining them with any other jobs in a work-sequence for a bus. If neither of nodes i, j is a long-duration job, include an arc (i , j) from node i to node j if a bus can handle job j after handling job i (the condition for including this arc is: (distance from q_i to p_j) $+ t^{\prime}_i \leq t_{-j}$). Let the resulting network be $G = (N, A)$ where N = set of nodes = { 1, 2, …, n }, and A = set of arcs.

By making small perturbations in the starting times of the jobs if necessary, we number the jobs serially in increasing order of their starting times. Then it is clear that if $i > j$, then (i , j) is not an element of A. This implies that the numbering of the nodes in G is an acyclic numbering (*i. e.* all arcs go from a node to another node with higher number). Hence G is an acyclic network, and it is the job precedence acyclic network for this set of jobs.

We define a simple chain in G to be either a set containing a single node or a sequence of more than one node, $i_1, i_2, \ldots, i_h$, satisfying the condition that (i_{g-1}, i_g) is an element of A for g = 2 to h. Thus it corresponds to the usual notion of a simple chain in network terminology when there are two or more nodes in it. However, a single node by itself is also considered as a simple chain (it has no arcs) in this context. Therefore, every work-sequence corresponds to a simple chain in G and *vice versa*.

Since each work-sequence corresponds to a simple chain in the acyclic network G and *vice versa*, the problem of partitioning the set of jobs into the smallest number of work-sequences is the same as that of finding a minimum cardinality simple chain cover for all the nodes in G, which is known in network programming literature as Dilworth's minimal chain decomposition problem. An efficient algorithm for it based on the maximum cardinality bipartite matching algorithm has been developed by Fulkerson (1956) and discussed in Ford and Fulkerson (1962) (see also Murty (1992)). The algorithm involves finding a maximum cardinality matching in the bipartite network B={N_1,N_2;A_1} with node set N_1={R_1, …,R_n}, N_2={C_1, …,C_n}, and edge set A_1={ (R_i,C_j): (i, j) is an arc in A in G}. Suppose the cardinality of a maximum cardinality matching in B in r. Then it is shown that the minimum number of simple chains needed to cover all the nodes in G is $s = n - r$; *i. e.* in our problem at least $n - r$ work-sequences or buses are needed to cover all the jobs. From any maximum cardinality matching M in B an easy procedure is available for deriving a set of $n - r = n - >|M|$ simple chains in G to cover all the nodes. This procedure consists of obtaining the set of arcs {(i, j): (R_i,C_j)} is an edge in the matching M} in G, it is the set of arcs in a node disjoint collection of simple chains in G, this collection of simple chains is a minimum cardinality simple chain cover for the nodes of G. Find it. The sequence of jobs corresponding to nodes in the order in which they appear on each of these simple chains is a work-sequence for a bus, and hence each of these simple chains can also be interpreted as a bus route.

12.5.2 Numerical Example

Job (or node) i is called a **predecessor** (or **ancestor**) of j if j can be handled by a bus after i, then we include the arc (i, j) in the job predecessor acyclic network G defined above. In addition, node i is called an **immediate predecessor** of j if it is a predecessor of j, and there is no other predecessor k of j for which i is also a predecessor.

Typically $\boldsymbol{G}$ will have too many arcs. So, in this $\boldsymbol{n}$ = 9 node example we show only a subnetwork $\boldsymbol{G}^{-}$ of $\boldsymbol{G}$, with an arc $(\boldsymbol{i}, \boldsymbol{j})$ in it only if $\boldsymbol{i}$ is an immediate predecessor of $\boldsymbol{j}$. With this, $\boldsymbol{j}$ can be handled by a bus after $\boldsymbol{i}$ if there is a simple chain (or directed path) from $\boldsymbol{i}$ to $\boldsymbol{j}$.

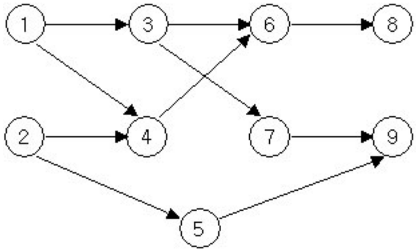


Figure 12.4. The subnetwork $\boldsymbol{G}^{-}$ for the example

So, after job 1, a bus can take up any of the jobs 3, 4, 6, 7, 8, or 9. The bipartite network $\boldsymbol{B}$ to apply Fulkerson's algorithm on this example is given in Figure 12.5.

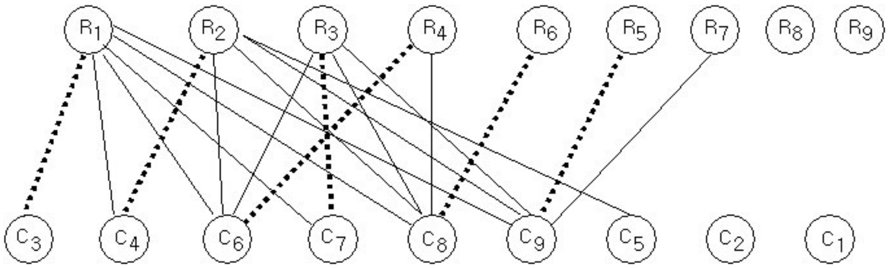


Figure 12.5. The bipartite network $\boldsymbol{B}$ for this example. A maximum cardinality matching in it is marked with dotted edges

$\boldsymbol{M}=\{(\boldsymbol{R_1},\boldsymbol{C_3}), (\boldsymbol{R_3},\boldsymbol{C_7}), (\boldsymbol{R_2},\boldsymbol{C_4}), (\boldsymbol{R_4},\boldsymbol{C_6}), (\boldsymbol{R_6},\boldsymbol{C_8}),(\boldsymbol{R_5},\boldsymbol{C_9})\}$ is a maximum cardinality matching in $\boldsymbol{B}$ with 6 edges. So, a minimal chain decomposition of $\boldsymbol{G}$ in this example has 9-6=3 chains, the arcs on these three chains are {(1,3), (3,7), (2,4), (4,6), (6,8), (5,9) }. Hence the three chains are {1,3,7}, {2,4,6,8},{5,9} with nodes appearing in the order listed, these are the work-sequences in a partition of the 9 jobs in this example into the smallest number of work-sequences.

This algorithm provides an efficient approach when **OBJ1** is the only objective to consider, ignoring the constraint on long duration work-sequences implied by **OBJ3**.

We will now discuss what happens if the maximum working time constraint in **OBJ3** is introduced. **OBJ3** is stated in the form of a goal of keeping the percentage of long duration work-sequences at 50 or less. Consider the simpler problem in which the set of jobs to be handled contains no long duration jobs, the driving time between the depots and any of the locations is 0, and it is required to partition this set of jobs into the smallest number of work-sequences each of which satisfies the maximum duration constraint of 12 hours. We will call a work-sequence satisfying this constraint a **feasible work-sequence**.

Define a new data element, l_{ij}, the length of arc (i, j) in G, to be $l_{ij} = t^{\text{f}}_j - t^{\text{f}}_i$. It is the total time lapse from ending time of job i to the ending time of job j that the driver of the bus has to spend if both job i and job j are assigned to the same bus. Let ζ denote a simple chain in G. If all the jobs corresponding to nodes on ζ are assigned to a bus on its route, then the time the driver of that bus will be working on that day is equal to length of ζ in G. So, a feasible work-sequence corresponds to a feasible simple chain which is a simple chain in G whose total length is less than or equal to 12 h and *vice versa*. So partitioning the set of jobs into the smallest number of feasible work-sequences is equivalent to the problem of finding a minimum cardinality simple chain cover for all nodes in G using only feasible simple chains.

We call this problem CSCP (for **constrained simple chain cover problem**). We now show that this problem is NP-hard.

Theorem 1 (Kim *et al.* 1999) CSCP, the problem of finding a minimum cardinality simple chain cover for all nodes in the acyclic network G with upper bound on the length of simple chains that can be used, is NP-hard.

Proof. We show that a well known NP-hard problem, a subset sum problem, is a special case of CSCP. Consider the subset sum problem (SSP):

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n = a_0$$
x_j is an element of $\{0,1\}$ for all j

where $a_1, a_2, \ldots, a_n$ are positive integers whose sum is an even number and $a_0 = (a_1 + a_2 + \ldots + a_n) / 2$
.

Because of the nature of the input data, if $x = (x_1, x_2, \ldots, x_n)^T$ is a feasible solution of SSP, then $y = (y_1, y_2, \ldots, y_n)^T$ where $y_j = 1 - x_j$, for all $j = 1$ to n, is also a feasible solution to it. We will call this solution y to be the complement of x.

Construct the acyclic network G' with $2(n+1)$ nodes as in Figure 12.6. The data by the side of each arc is its length.

Consider the problem, CSCP, of finding a simple chain cover for nodes in G' using only feasible simple chains (those with length $\leq a_0$).

We will show that this CSCP has a solution using only two feasible simple chains if and only if SSP has a feasible solution. Suppose a solution to this CSCP uses only two feasible simple chains called Chain 1 and Chain 2. We will represent each of these chains by the sequence of nodes on it. Let Chain 1 be $u_1, u_2, \ldots, u_n$, and Chain 2 be $v_1, v_2, \ldots, v_n$.

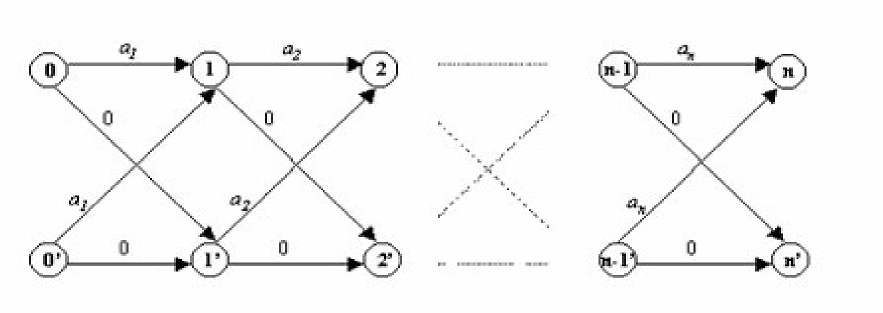


Figure 12.6. Acyclic network ***G'***

For each j = 0 to n since nodes j and j' are not related in G' (*i. e.* there is no chain from j to j' or from j' to j), the nodes j and j' must be contained on different chains among Chain 1 and Chain 2. So, if $u_j = j$, v_j must be j' and *vice versa* for all j = 0 to n.

For $j = 0$ to n **- 1**

If $u_j = j$ or j' and $u_{j+1} = j + 1$, in this case $v_j = j'$ or j, respectively, and $v_{j+1} = (j + 1)'$, define x_{j+1} to be = 1 and y_{j+1} to be = 0.

If $u_j = j$ or j' and $u_{j+1} = (j + 1)'$, in this case $v_j = j'$ or j, respectively, $v_{j+1} = (j + 1)$, define x_{j+1} to be = 0 and y_{j+1} to be = 1.

It can be verified that vectors $x = (x_1, x_2, \ldots, x_n)^T$ and $y = (y_1, y_2, \ldots, y_n)^T$ obtained in this way are both feasible solutions of SSP.

Conversely, given any feasible solution $x = (x_1, x_2, \ldots, x_n)^T$ for SSP, let y be its complement.

Using x and y, and the correspondences defined above, we can obtain a solution to this CSCP having only two feasible simple chains in the cover. This establishes a one-to-one correspondence between feasible solutions of SSP and simple chain covers for this CSCP using only two feasible simple chains.

Since SSP is known to be NP-hard, the problem of checking whether our CSCP has a cover with only two feasible simple chains is also NP-hard. Therefore the problem of finding a minimum cardinality simple chain cover in an acyclic network using only feasible simple chains is also NP-hard. □

The CSCP problem remains NP-hard even if the network G has special properties like bipartiteness. To show this, consider the acyclic bipartite network $G'' = (N_1, N_2; A'')$ where $N_1 = \{1,2,\ldots,n\}$, $N_2 = \{1',2',\ldots,n'\}$, and $A'' = \{(i,i')$ with weight a_1 for i = to n } U $\{(i,i')$ i = 1 to $n, j > i$ with weight 0}. Then it can be shown that the SSP has a feasible solution if and only if there exists a simple chain cover for nodes in G'' with exactly two feasible simple chains (those with weight $\leq a_0$).

Because of this, we have decided to ignore **OBJ3** for determining the initial set of work-sequences, and handle **OBJ3** afterwards heuristically, interactively. This will be discussed in Section 12.5.4.

12.5.3 How to Handle OBJ2?

OBJ2 can be split into two parts as: **OBJ2** = **OBJ2.1** + **OBJ2.2**, where **OBJ2.1** = cost of empty-load drives between consecutive jobs on all the work-sequences used, **OBJ2.2** = cost of empty-load drives from the depot to the starting location of the first job in the work-sequence, and from the ending location of the last job in the work-sequence to the depot, for all the buses used.

It can be verified that **OBJ2.1** is uniquely determined by the set of work-sequences into which the jobs are partitioned, and does not depend at all on the depot from which a bus is allotted to each of the work-sequences adopted. Likewise OBJ2.2 mainly depends on the depots from which buses are allotted to work-sequences, and not so much on how the work-sequences are formed.

So, we handle **OBJ2** by optimizing its two parts at separate stages of the algorithm. Next we show how we handle **OBJ1**, **OBJ2.1** together to determine the initial set of work-sequences. Minimizing **OBJ2.2** is taken as the objective for determining from which depot to allot a bus to each work-sequence, after the set of work-sequences to adopt is finalized.

12.5.4 Initial Set of Work-sequences

The strategy solves

> **Problem 1**: Find s = minimum number of work-sequences into which the set of jobs {1,...,n} can be partitioned as discussed in Section 12.5.1. There may be several such minimal partitions, among all of them find the one that has minimum value for **OBJ2.1**.

The arc $(\boldsymbol{i}, \boldsymbol{j})$ in the network $\boldsymbol{G}$ represents the opportunity of a bus servicing job $\boldsymbol{j}$ after servicing job $\boldsymbol{i}$. If this happens, then that bus travels from location $\boldsymbol{q_i}$ after finishing the servicing of job $\boldsymbol{i}$ at clock time $\boldsymbol{t^-_i}$, to location $\boldsymbol{p_j}$ to start the servicing of job $\boldsymbol{j}$ at clock time $\boldsymbol{t_{-j}}$.

Then in the time interval between $\boldsymbol{t^-_i}$ to $\boldsymbol{t_{-j}}$ of length $\boldsymbol{t_{-j}} - \boldsymbol{t^-_i}$ hours representing the empty-load drive corresponding to this arc $(\boldsymbol{i}, \boldsymbol{j})$ the bus and the driver are working but the company gets no profit from it. This time has been estimated to cost at the rate of \$40/hour. Hence we take the cost $\boldsymbol{c_{ij}}$ (empty-load drive cost) of arc $(\boldsymbol{i}, \boldsymbol{j})$ in $\boldsymbol{G}$ to be \$40$(\boldsymbol{t_{-j}} - \boldsymbol{t^-_i})$; this is the cost coefficient of arc $(\boldsymbol{i}, \boldsymbol{j})$ in $\boldsymbol{G}$ and the corresponding arc $(\boldsymbol{R_i},\boldsymbol{C_j})$, in $\boldsymbol{B}$ for OBJ2.1.

The maximum cardinality matching problem in $\boldsymbol{B}$ usually has many alternate optimum solutions, and any one of them can be used to get a minimum cardinality simple chain cover for the nodes in $\boldsymbol{G}$. So, in order to partition the set of jobs into work-sequences minimizing **OBJ1**, **OBJ2.1** simultaneously, we need to find a minimum cost maximum cardinality matching in the bipartite network $\boldsymbol{B}$ with $(\boldsymbol{c_{ij}})$ as the vector of arc cost coefficients. Since $\boldsymbol{B}$ is bipartite, this can be found very efficiently using the min cost max flow algorithm. If $\boldsymbol{M}^{-}$ is a min cost max cardinality matching in $\boldsymbol{B}$, the set of arcs $\{(\boldsymbol{i},\boldsymbol{j})$: $(\boldsymbol{R}_i,\mathrm{C}_j)$ is a matching arc in $\boldsymbol{M}$-$\}$ is a node disjoint collection of simple chains in $\boldsymbol{G}$. The collection of work-sequences corresponding to these simple chains minimizes **OBJ2.1** among all partitions of the set of jobs into the smallest possible number of work-sequences.

This set of work-sequences is a solution for **Problem 1**, it is the partition of the jobs {1,...,n} into an initial set of work-sequences to be considered in the heuristic approach for handling **OBJ3** in the next section.

12.5.5 How to Handle OBJ3?

The partition of jobs {1,...,n} into a set of work-sequences obtained above considered only optimizing **OBJ1** and **OBJ2.1** in this priority order, but totally ignored **OBJ3**. Now we will discuss how to modify this partition taking OBJ3 into account. Define for a work-sequence its **total working time** = the difference between the ending time of the last job and the starting time of the first job on the work-sequence expressed in hours.

The duration of a work-sequence defined earlier depends on the depot from which the bus for this work-sequence is allotted, it is equal to its total working time plus the distance from and to the depot. Giving an allowance of one hour for driving from and to the depot, we call work-sequences for which the total working time is greater than or equal to 11 h (= safety time −1) as **long work-sequences**, and these are the work-sequences most likely to violate the maximum duration constraint.

These long work-sequences are of two types. The first type is the single job work-sequences that consist of just one long-duration job. The second type is multiple job work-sequences that are long. Typically, less than 10% customer service requests are long duration jobs. The company likes these because they generate higher fees, and they try to assign them with equal frequency among all their drivers. Long-duration jobs almost always contain nondriving rest periods during which the driver can take a nap and get refreshed. For this reason single-job long work-sequences are never considered a problem.

OBJ3 is a goal requiring that the percentage of long duration work-sequences should be ≤ 50 as far as possible. When this goal is violated, some of the multiple-job long duration work-sequences are modified using the following heuristic strategies:

On each of these long multiple job work-sequences, the longest arc (*i. e.* an arc $(\boldsymbol{i},\boldsymbol{j})$ on this work-sequence having maximum $\boldsymbol{l}_{ij}$ value) is deleted from the network $\boldsymbol{G}$, and the algorithm discussed above applied on the remaining network. Almost always this produces a new collection of work-sequences that satisfies the goal on the percentage of long work-sequences while increasing the number of work-sequences only slightly. Otherwise this process is repeated.

Clearly the optimum values of **OBJ1**, **OBJ2.1** obtained in the first run of the algorithm are lower bounds for the minimum values of these respective objective functions in the specified priority order for partitioning the set of jobs considered into work-sequences while satisfying the goal in **OBJ3**. We use these lower bounds to compare the quality of the final solutions obtained.

12.6 The Model to Allocate a Bus to Each Work-sequence

After finalizing the partition of the set of jobs into work-sequences, we turn to the problem of assigning buses to these work-sequences. These buses can come either from depot 1,2, or outside vendors. The assignment of buses to work-sequences will be carried out so as to minimize:

OBJ2.2 (the cost of empty-load drives from the depot to the starting location of the first job, and from the ending location of last the job to the depot, for all the company's own buses used) + the rental cost of buses from outside vendors that are used that includes besides **OBJ2.2** defined in Section 12.5.2 the rental cost and the cost of such empty-load drives of buses rented from outside vendors.

Let

$\boldsymbol{p}$ = number of work-sequences in the final set,
$\boldsymbol{c_t}$, $\boldsymbol{d_t}$ = cost of the empty-load drive at the beginning and at the end of the $\boldsymbol{t^{th}}$ work-sequence if a bus is assigned to it from depot 1 and 2 respectively,
$\boldsymbol{e_t}$ = cost of renting a bus for the $\boldsymbol{t^{th}}$ work-sequence from an outside vendor,
$\boldsymbol{N_1}$, $\boldsymbol{N_2}$ = number of buses available at depot 1 and 2, respectively.

Since $\boldsymbol{e_t}$ is typically much larger than $\boldsymbol{c_t}$ or $\boldsymbol{d_t}$ the number of buses rented from outsider vendors will be $(\boldsymbol{p} - \boldsymbol{N_1} - \boldsymbol{N_2})^+ = \max\{0, \boldsymbol{p} - \boldsymbol{N_1} - \boldsymbol{N_2}\}$. We have three sources for buses, sources 1, 2, and 3 (these are depots 1, 2, and outside vendors respectively), with availability of buses equal to $\boldsymbol{N_1}$, $\boldsymbol{N_2}$, and $(\boldsymbol{p} - \boldsymbol{N_1} - \boldsymbol{N_2})^+$ respectively. Each work-sequence requires exactly one bus. Clearly the problem of assigning buses to work-sequence can be modeled as a (3 x $\boldsymbol{p}$) transportation problem, (TP1), with the (3 x $\boldsymbol{p}$) cost matrix whose $\boldsymbol{t^{th}}$ column is $(\boldsymbol{c_t},\boldsymbol{d_t},\boldsymbol{e_t})^{\mathrm{T}}$ for $\boldsymbol{t}$ = 1 to $\boldsymbol{p}$.

12.7 Numerical Results

Results obtained by using the i-DMSS on requested job data at the company over a 5 day period from the past are shown in Table 12.1.

The second column in Table 12.1 gives the number of small group jobs on the day. The third and fourth columns give values of **OBJ1** and **OBJ2.1** respectively in the set of initial work-sequences obtained for small group jobs. They are the lower bounds for these objectives for small group jobs. The fifth, sixth, and seventh columns give the final values for **OBJ1**, **OBJ2.1**, and **OBJ2.2**, respectively, for small group jobs. The eighth column gives the number of remaining jobs for which 45-seat buses are to be allotted. The ninth and tenth columns give the values for **OBJ1** and **OBJ2.1** in the initial set of work-sequences obtained for these jobs. They are the lower bounds for **OBJ1** and **OBJ2.1**, respectively, for handling these jobs. The eleventh, twelfth, and thirteenth columns give the final values for **OBJ1**, **OBJ2.1**, and **OBJ2.2**.

Table 12.1. Numerical results of the i-DMSS

Day	Small group jobs						Large group jobs					
	# of jobs	Omit Obj3		Meet OBJ3			# of jobs	Omit Obj3		Meet OBJ3		
		OBJ1	OBJ2.1	OBJ1	OBJ2.1	OBJ2.2		OBJ1	OBJ-2.1	OB-J1	OBJ-2.1	OBJ-2.2
1	14	5	3070	5	3070	2170	54	22	8100	23	9520	7700
2	13	5	2650	5	2650	2480	43	17	8490	17	8810	7540
3	12	5	2320	5	2320	1980	44	18	9980	19	11800	7480
4	14	5	3600	5	3600	2200	51	21	9150	21	9630	8400
5	15	4	2300	4	2300	1800	55	23	9570	24	9810	8330

In Table 12.1, the final values for **OBJ1** and **OBJ2.1** for the small group jobs are the same as the lower bound for these respective objective functions for this subset of jobs.

To meet **OBJ3**, for large group jobs the value of **OBJ1** increased by at most 1 over its lower bound on three of the five days while **OBJ2.1** increased about 9.5% on average over its lower bound. This shows that the final solutions obtained by the i-DMSS had objective values quite close to the lower bounds for these objective functions.

We found that the results obtained by using the i-DMSS on this data are between 10-20% more economical than the manual decisions made by the senior employees for those days.

12.8 Conclusions

We described how we used relaxations, hierarchical decomposition, and heuristics to model and analyze the complex problem of allotting buses at a bus rental company, and presented the numerical results obtained with this approach. The new person responsible for making the allocation decisions at the company uses the DMSS in an interactive manner to make all the decisions in about a couple of hours every evening.

Appendix

The following data on 57 jobs requested at the chartered bus company on one day is given as an illustrative example. Figure 12.7 is the travel-time matrix and Figure 12.8 gives the other data for these jobs.

from · to	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	0	85	45	60	80	25	30	30	55	65	95	110	110	55	35	105	85	65	50	50	85	35
2	85	0	85	55	35	65	105	105	75	40	40	25	30	30	60	40	55	75	105	40	20	65
3	45	85	0	35	60	35	35	75	90	85	110	110	100	60	65	80	55	25	25	50	95	25
4	60	55	35	0	25	40	65	85	85	70	90	80	65	40	60	45	25	20	55	25	70	25
5	80	35	60	25	0	55	85	100	90	65	70	60	40	35	65	25	20	40	80	30	55	45
6	25	65	35	40	55	0	40	50	55	55	80	90	90	35	30	80	65	45	50	30	70	15
7	30	105	35	65	85	40	0	45	85	90	120	130	125	75	60	110	85	60	25	65	110	40
8	30	105	75	85	100	50	45	0	50	70	105	125	135	75	45	125	110	90	70	70	100	60
9	55	75	90	85	90	55	85	50	0	35	60	85	105	55	25	110	105	95	105	60	60	65
10	65	40	85	70	65	55	90	70	35	0	35	55	70	30	30	80	85	85	100	45	30	60
11	95	40	110	90	70	80	120	105	60	35	0	30	60	50	60	80	90	105	130	65	20	90
12	110	25	110	80	60	90	130	125	85	55	30	0	30	55	80	55	80	100	130	65	25	90
13	110	30	100	65	40	90	125	135	105	70	60	30	0	60	90	30	55	80	120	65	50	85
14	55	30	60	40	35	35	75	75	55	30	50	55	60	0	35	55	55	60	80	15	35	40
15	35	60	65	60	65	30	60	45	25	30	60	80	90	35	0	90	80	70	80	35	55	40
16	105	40	80	45	25	80	110	125	110	80	80	55	30	55	90	0	30	60	100	55	60	70
17	85	55	55	25	20	65	85	110	105	85	90	80	55	55	80	30	0	30	75	45	75	50
18	65	75	25	20	40	45	60	90	95	85	105	100	80	60	70	60	30	0	45	45	90	30
19	50	105	25	55	80	50	25	70	105	100	130	130	120	80	80	100	75	45	0	70	115	45
20	50	40	50	25	30	30	65	70	60	45	65	65	65	15	35	55	45	45	70	0	50	25
21	85	20	95	70	55	70	110	100	60	30	20	25	50	35	55	60	75	90	115	50	0	70
22	35	65	25	25	45	15	40	60	65	60	90	90	85	40	40	70	50	30	45	25	70	0

1 to 20 are the numbers for various locations which are starting/ending locations of jobs.
21 and 22 are the numbers of northern depot and southern depot, respectively.

Figure 12.7. Sample travel time(in minutes of driving)

Job	A	B	C	D	E	Job	A	B	C	D	E	Job	A	B	C	D	E
1	11	5:00	19	8:50	2	2	19	5:00	19	16:50	2	3	14	5:30	5	8:30	2
4	14	6:00	5	18:50	2	5	17	6:00	16	9:50	1	6	7	6:00	12	10:50	2
7	14	6:00	13	18:30	2	8	10	6:30	18	11:50	2	9	15	6:50	20	12:00	2
10	2	7:00	13	14:00	1	11	5	7:00	2	11:30	2	12	17	7:30	8	9:00	2
13	5	7:30	4	9:30	2	14	6	7:30	16	18:50	2	15	6	7:50	5	19:50	2
16	10	7:50	7	20:30	2	17	14	8:00	4	18:50	2	18	1	8:30	4	10:30	2
19	20	8:30	15	11:30	2	20	3	8:50	13	21:30	1	21	9	8:50	17	10:30	1
22	19	8:50	8	10:30	1	23	3	9:00	2	16:30	2	24	4	9:00	10	16:00	2
25	8	9:30	14	12:00	2	26	13	9:30	5	20:30	2	27	9	9:30	9	18:50	1
28	2	9:30	2	16:50	1	29	10	10:30	5	21:00	2	30	16	10:30	4	16:00	2
31	15	10:50	1	18:30	1	32	14	11:00	11	13:50	1	33	16	12:50	14	15:30	1
34	10	12:50	14	15:50	2	35	12	13:00	5	14:00	2	36	3	13:50	6	21:30	1
37	14	14:30	16	20:00	2	38	14	15:50	19	20:00	2	39	18	16:00	15	20:00	2
40	6	16:30	4	18:00	2	41	12	16:50	1	23:00	2	42	2	17:00	4	21:00	2
43	19	17:30	11	21:50	2	44	13	17:30	15	23:50	2	45	18	18:00	14	21:00	1
46	7	18:30	1	22:50	2	47	13	18:30	11	20:30	2	48	5	18:50	6	23:50	2
49	16	18:50	17	21:30	1	50	5	18:50	8	19:50	2	51	18	19:30	2	21:30	2
52	18	19:50	14	21:30	1	53	15	19:50	16	23:50	2	54	8	20:00	16	22:50	2
55	19	20:30	10	21:50	2	56	1	21:50	11	22:50	2	57	12	21:50	4	23:50	2
58	13	22:30	3	23:50	2	59	4	22:30	18	23:50	2						

A:Starting location *B*:Starting time *C*:Ending location *D*:Ending time *E*:Bus type
(1 is 15 seat bus, 2 is 45 seat bus.)

Figure 12.8. Requested jobs on one day

References

Ackoff, R (1996) The future of operational research is past. Journal of the Operational Research Society 30(2):93-104

Ford LR and Fulkerson DR (1962) Flows in Networks. Princeton University Press, Princeton.

Fulkerson DR (1956) Note on Dilworth's decomposition theorem for partially ordered sets. Proceedings of American Mathematical Society 7: 701-702

Geoffrion AM (1976) The purpose of mathematical programming is insight, not numbers. Interfaces 7 1(1):81-93

Kim WJ, Kabadi SN, and Murty KG (1999) A chartered bus allocation and scheduling problem. Technical Report, Department of IOE, University of Michigan, Ann Arbor

Murty KG (1992) Network Programming. Prentice Hall: Englewood Cliffs, New Jersey.

Murty KG (2005 forthcoming) Optimization Models for Decision-making. Volume 1, Junior level web-book, (under preparation)

Murty KG,d Djang PA (1999) The U.S. Army national guard's mobile training simulators location and routing problem. Operations Research 47(2):175-182.

Simon HA (1987) Two heads are better than one: the collaboration between AI and OR. Interfaces 17:8-15

10. Wolfram S (2002) New Kind of Sciences. Wolfram Media, UK.

13

MicroDEMON: A Decision-making Intelligent Assistant for Mobile Business

Guliano Pistolesi

ThinkinGolem SC, Piazza Prati degli Strozzi 21, 00195 Rome, Italy,
e-mail: guliano.pistolesi@thinkingolem.com

The recent massive use of wireless technology in the business domain strongly modified organization and management of work and made it critical both to gather decision-problem data and to share them among human business agents in real-time. To support this new generation of decision makers and face real-time in-the-field decision problems, we developed MicroDEMON, a language-based user-centered mobile software system that represents a step further along the evolution of active DMSSs, a kind of intelligent and proactive decision-making support systems.

13.1 Introduction

Active DMSSs (ADMSSs) are a kind of intelligent decision-making support systems that take the initiative without getting specific orders. They respond to nonstandard requests and commands when dealing with tasks in problem solving that are ambiguous and/or complex. They are expressions of an agent-based approach to DMSS in order to enhance creativity in collaborative human-computer problem solving (Angehrn 1993). In a previous study we pointed out the importance of using agents equipped with natural language-processing skills as intelligent interfaces for DMSSs (Pistolesi 2002). This represents a step further along the evolution of ADMSSs since language becomes the mean to make more natural and user-friendly human-computer interaction (Maes 1998). In decision problem solving this leads to a sort of artificial brainstorming analogous to that among human decision makers: the intelligent agent meets the decision-maker cognitive style to give rise to a group decision.

The recent massive use of wireless technology (*e.g.* mobile phones, smartphones, palmtops and PDAs) in the business domain strongly modified the organization and management of work (Perry *et al.* 2001), and made it critical both to gather decision problem data and to share them among human business agents in real-time (Pica and Kakihara 2003). This appeared to be a suitable domain where to apply our new kind of ADMSS, since today mobile workers represent a significant portion of a company's overall staff and they are often responsible for making critical business decisions.

To help this new generation of decision makers, we developed MicroDEMON, a J2ME application for mobile business that implemented the new ADMSS architecture outlined in Pistolesi (2002). Our application, which can be installed on any mobile device equipped with a Java Virtual Machine (*e. g.* Nokia), includes both a versatile and suitable multicriteria decision-making model, namely, AHP (Saaty 1990), and an intelligent interface agent that helps, by natural-language interaction the human mobile worker to manage complexity of the decision model adopted.

Another strongpoint of our approach is that MicroDEMON can work not only as a mobile standalone ADMSS, but also as a node of a global network of agents, connected together to the company headquarters to gather or access business decision problem data stored on an Internet repository: when a MicroDEMON installed on a mobile device must face a new decision problem together with the human mobile worker it was assigned to, it sends a request to the company's repository on the Web looking for similar decision problems that could help either finding an effective solution or taking a decision in less time and accordingly to the company's purposes and scope; in any case, when the decision problem is solved by the strange team composed of the mobile worker plus the intelligent agent, data about it are sent to the repository to be integrated in the database and then shared with other MicroDEMONs and the corresponding human business operators.

With respect to most of the few current mobile business DMSS, which are company-centered and not truly distributed (San Pedro *et al.* 2003), MicroDEMON is a user-centered application that enables real-time group decision-making at different levels, allowing for effective management of global enterprise decision complexity by mobile technology.

The main purpose of this chapter is to provide an overview of ADMSSs and decision making in mobile business, then to describe the MicroDEMON decision support system to show how useful and promising is the adoption of natural language-processing agents as the building blocks of ADMSSs for decision-making support.

13.2 Active Decision-making Support Systems

There is no universally accepted definition for a DMSS, as the substance of a support system is geared to the context for which it has been developed, to available data, information and knowledge, and to the users using the system. There are, however, some elements and some characteristics commonly accepted and recognized as being parts of a DMSS: A DMSS is, in most cases, an interactive, computer-based system, which both has functions for giving aid to users in judgment and choice activities, by means of a supportive and easy-to-use interface, and represents the necessary and sufficient knowledge about a decision problem (Er 1988, Gorry and Scott-Morton 1971, Sprague 1980, Turban and Aronson 1998). It provides data storage and retrieval, support framing, modeling, and problem solving at various levels of an organization. Typical application areas of DMSSs are management and planning in business, health care, military, and any area in which management faces complex decision situations.

While the quality and reliability of modeling tools and the internal architecture of a DMSS are important, the most crucial aspect of such a system is, by far, its user interface (Angehrn 1993). Current issues of a DMSS can provide several interface facilities using friendly concepts and properties of a graphical user interface (GUI), following the *direct manipulation* metaphor (Schneiderman 1988). However, this approach could be a somewhat hit-and-miss affair in practice, requiring the user to apply significant management skills, while, on the other hand, the *personal assistant* metaphor requires that the user only be confident with a more reliable intelligent supporting system (Chin 1998). ADMSSs represent a recent and evolving research paradigm in the DMSSs arena that makes real intelligent assistant agents meeting decision support systems to help the user in a more effective way.

Actually, the classical DMSS paradigm implies to provide only a powerful set of tools and constructs for problem representation, operations on data, memory aids to assist users in decision modeling and mechanisms for control (Sprague 1980). Those passive systems are totally under explicit control of the decision maker and they do not provide any intelligent assistance for problem solving. Then, a natural evolution of DMSSs was the inclusion of intelligent features to support the decision maker in the various phases of the decision-making process as identified by Mintzberg *et al.* (1976).

Turban and Aronson (1998) provided basic discussions on the potential synergy between research in artificial intelligence (AI) and the DMSSs domain. Initial approaches to include AI in a DMSS focused both on providing an expert assistance in strategy formulation and on using an expert system component to assist the user during the problem identification stage of decision making (Edwards *et al.* 2000). Keen (1987) addressed the need for a more active kind of decision support technology in the decision-making process, namely, intelligent agents.

Intelligent agents are software systems that inhabit some virtual, complex, dynamic environment, sense and act autonomously in this environment to realize a set of goals or tasks for which they are designed (Maes 1994). Recently, intelligent agents have been proposed as one mechanism to help computer users in dealing with work and information overload (Norman 1998). The selling points for these programs are that they claim to have captured the essential qualities of human intelligence, that means reasoning, problem solving, learning, and other qualities apparently central to the capacity we call intelligence. The agent technology could be used to engage and help all types of end users with such tasks as information filtering, information retrieval, mail management, meeting scheduling, selection of books, movies, music, and so forth (Norman 1994). Agents that can build up goals and perform tasks driven by those goals are certainly useful in several contexts (Elliott and Brzenzinski 1998). They are able to understand the users' needs and adapt the software system interface to make the human-computer interaction more effective in performing a task. Thus, agents might reasonably be components of an ADMSS.

This new type of focus contributed to the development of ADMSSs based on intelligent agents technology and advocates the development and implementation of DMSSs that encompass a set of tools that actively participate in the various phases of the decision-making process. The system component that provides this assistance is referred to as a Computer-directed Process Manager (CDPM).

Current approaches to implementation of autonomous processes use daemons as intelligent agents which monitor the task-related events of the user and, based on those events, trigger appropriate behaviors. Raghavan (1991) proposed that those behaviors should include support for coordination of the decision process. The types of behaviors that autonomous process would demonstrate include as observing a decision-making process and scheduling the necessary related tasks as keeping track of the pending tasks and enforcing constraints. These types of behaviors coincide with the requirements for supporting the control phase of coordinating organizational decision-making processes.

13.3 Decision-making in Mobile Business

Mobile applications technology enables deep structural change in how organizations accomplish goals by cost-effectively moving existing automated business processes beyond organizations' premises to wherever and whenever those processes are engaged most efficiently (Scheer 1998). Mobile technology allows organizations to adopt new kinds of service delivery and interaction, culminating in significant productivity gains. Mobile business extends the robust business intelligence environment of the office to the mobile community. Today, the mobile workforce requires more and more up-to-date information in order both to make the right decisions and to offer the highest level of service to customers and partners.

The rapid and accelerating move towards mobile technology has increasingly provided people and organizations with the ability to work away from the office and on the move. Decision makers in enterprises have been the primary target group for new information and communication technology due to their capability to adopt and invest on ICT equipment and services. As organizations are increasingly introducing personal mobile devices to support work, the need for understanding the dynamics between mobile devices, organizations and individual's work assumes a privileged position in information systems research. In fact, both academic and industry research efforts are focusing on maximizing the interaction between individuals and their mobile devices. This is based on the idea that personal mobile devices will bring about fluid work organizations to sustain a postmodern era of efficient and *ad hoc* services available "anytime, anywhere" through the increased mobility and connectivity of professionals. For this reason, planning and implementing a mobile-based enterprise strategy could be puzzling (Tyrvainen and Veijlainen 2003).

Mobile users making real-time decisions face various types of uncertainties in the decision environment (San Pedro *et al.* 2003). Mainly, information held in a mobile device is likely to be incomplete or outdated and may not reliably support user's needs in critical situations. Users need an effective support when they face critical situations in-the-field and would welcome alerts about the reliability of data in such situations. Various multicriteria decision-making approaches (Keeney and Raiffa 1976) can be incorporated into a handheld device to resolve conflicts among many factors influencing the choice of the best alternative when making decisions in a mobile business environment.

Mobile workers have typically to face decisions that can be organized according to the Gorry and Scott-Morton (1971) framework. Access to accurate, up-to-date information accelerates decision making. It boosts efficiency, improves consistency and reduces duplication of effort. In the end, this contributes to greater customer satisfaction. The challenge for managers today is to provide real-time access to information when over one third of the corporate workforce is spending increasing amounts of time out of the office. Mobile workers represent a significant portion of a company's overall staff, and these workers are often responsible for making critical decisions or providing specific expertise. Their needs are priorities that must be met. Mobile workers require not only access but also versatility, in the ways they manage time and communicate with existing and potential customers, partners and vendors, and colleagues at branch or head offices. They need to achieve results and maintain a healthy work/life balance, at the same time.

Studies on mobile workers highlighted different facets of access to remote people and information, and different facets of the *anytime, anywhere* concept. Four key factors in mobile work were identified (Perry *et al.* 2001):

- the role of planning;
- the chance to work during *dead times*;
- the access to remote technological and informational enterprise resources;
- the chance to monitor and get feedback on the activities of remote colleagues.

These issues were used to explore key notions of resource and task flexibility: how mobile workers make effective use of their time, how they maintain awareness about remote activities, and how they are able to access remote information and devices. This study outlined properties that should characterize the technology that would support the behavior of mobile workers (Perry *et al.* 2001):

- due to the intrinsic unpredictability of the in-the-field working environment, mobile systems should be highly flexible, rather than highly specific, since the ability to adapt to a great variety of situations would be more useful than highly complex and powerful, but single-use, systems;
- to allow more effective planning of activities and flexible organization of work for workers coordination while mobile;
- to support effective use of the dead time to plan for upcoming mobile activities;
- to allow the location, use of, and access to locally available resources or location-based services;
- to allow monitoring of remote activities more easily, perhaps through the use of subscriptions to text-based office information.

All of those properties characterize the MicroDEMON software system and make it an appropriate technological solution to support mobile workers in the decision-making process.

13.4 MicroDEMON As A Decision-making Intelligent Assistant for Mobile Business

Till now, decision-making support systems developed for mobile business were enterprise-centered, with the organization service providers able to understand the user's context as well as to communicate their local context to the user to gain competitive advantage over other companies. For example, online special offers can adapt to the mobile-worker's location, time of day, or special circumstances that can make the service valuable to customers. An enterprise service provider can incorporate a dynamic business model that can adapt itself according to its capability to provide e-service, to the user's needs and the system's capability to handle user's requests, and to respond to those requests.

However, because of the complexity and uncertainty in mobile computing due to the great instability in network connection which can black-out during activities, there is a strong need to provide a DMSS able to support real-time decision-making in-the-field when disconnected: a user needs to make informed decisions especially during critical situations, then the right approach is to decentralize the support system and to distribute the whole architecture of a DMSS along the mobile network.

MicroDEMON was developed to accomplish this aim, that is, to shift the enterprise-centered approach to mobile worker decision support to a more effective user-centred approach. Our ADMSS is equipped with:

- a flexible multicriteria decision model (*i. e.* AHP, Saaty 1990), to adapt to a changing decision environment;
- a natural language-based intelligent assistant that represents a light issue of the DEMON interface agent outlined in Pistolesi (2002) to help the user in dealing with the decision problem complexity.
- a network communication system to allow the user to send queries/requests about previously solved decision problems data that meet the current environment features so as to increase decision-making effectiveness with the enterprise expertise;
- a storing system to save user preferences about the decision-making workflow (*e. g.* sequence of how AHP tasks are accomplished, evaluation scale type, and so on) and support evolutionary usage.

So as to have mobile workers relying on:

- a growing repository where faced decision problems are stored in the XML information format and can be read/modified by office decision makers to better fit the enterprise decision objectives and strategies;
- a group decision-making system to share XML-structured information about decision problems with other colleagues to enhance group expertise in decision-making.

The whole system gives to users ubiquitous access to the information they need to make real-time decisions based on the set of enterprise data. Gorry and Scott Morton (1971) proposed that the attributes of information (*e. g.* accuracy, source, aggregation level, currency) vary depending on the level of managerial activities (Anthony 1965) and relative degree of structure in the decision-making (Simon 1960). Many other researchers suggested that information processing requirements depends on task uncertainty, mechanism for control and coordination, the nature of decision making, and the level of management activities (Alloway and Quillard 1983, Davis and Olson 1985, Gorry and Scott-Morton 1971). In general, what these researchers emphasized is that a DMSS should be designed to provide an appropriate amount and quality of information according to task variety, in order that users may clarify ambiguities and define problems (Daft and Lengel 1986; Rice 1992). Since the most important result of a session with a DMSS is insight into the decision problem, to accomplish this goal a good user interface to a DMSS supports model construction and model analysis, reasoning about the problem structure in addition to numerical calculations and both choice and optimization of decision variables.

MicroDEMON provides all the functions of an ideal DMSS and manages interaction with the user by means of a natural language-based friendly interface, nearer to the human interaction and reasoning style (Bobrow 1968). It is able to vary the amount and quality of information provided (Barnard *et al.* 1981) and to manage individual differences and cognitive styles in problem-solving and decision-making (Cowan 1991, Hayes and Allinson 1994, Nutt 1990). The system learns about the user and adapts to him/her, and, on the other hand, the user learns about the DMSS and adapts his behavior accordingly, since a truly adaptive system should support the user's learning of the system function (Sasse 1992). The agent is an active participant in the interaction (Carroll and Campbell 1989), which can decide and manage the course of interaction and not just follow embedded decisions made at design time (*i. e.* normative decisions). However, such a support system is able to negotiate its decisions with the learner and not just impose them, since in pursuing its decision-making supporting goals it relies explicitly on the relationship established with the user (Vassileva 1996).

Then, the central idea of this mobile system is that the machine and the user together form a team, able to give rise to a sort of artificial brainstorming between members in a group-decision process, with the system providing the kind of active decision support advocated by Raghavan (1991). Thus, this mode of working is completely flexible and it allows the user's skills to complement the ability of the machine in providing powerful support for accomplishing effective decision making in this complex domain.

13.4.1 The MicroDEMON Architecture

Whereas a wide variety of DMSSs exists, such a system is typically made of three fundamental components (Sage 1991):

- the Data-base management system (DBMS), which is used as a data bank to store large quantities of data that are relevant to the class of problems for which the DMSS has been designed; moreover, it provides logical data structures with which the user interacts;

- the Model-base management system (MBMS), which has as primary function to transform data from the DBMS into information that is useful in decision-making by assisting the user in model building and in framing unstructured decision problems;
- the Dialog generation and management system (DGMS), which should be an intuitive and easy-to-use interface to aid the decision maker in model building to maximize benefits from the system support (Preece 1994);

The architecture of MicroDEMON, implemented in J2ME, that is, Java for microdevices (*e. g.* mobile phones, smartphones, palmtops, PDAs, and so on), follows this schema (see Figure 13.1), with the exception that the DGMS component, which is an interface agent, coincides with the CDPM element of ADMSSs (Sprague 1980). An interface agent differs from an ordinary interface since it is expected to change its behaviors and actions autonomously according to the human user's behaviors as the interaction progresses. Since Weizenbaum's program ELIZA (Weizenbaum 1976) natural language-based interface agents were successfully used to make more effective human-computer interaction by means of a more natural way for humans to interact with an artificial software system (Maes 1998).

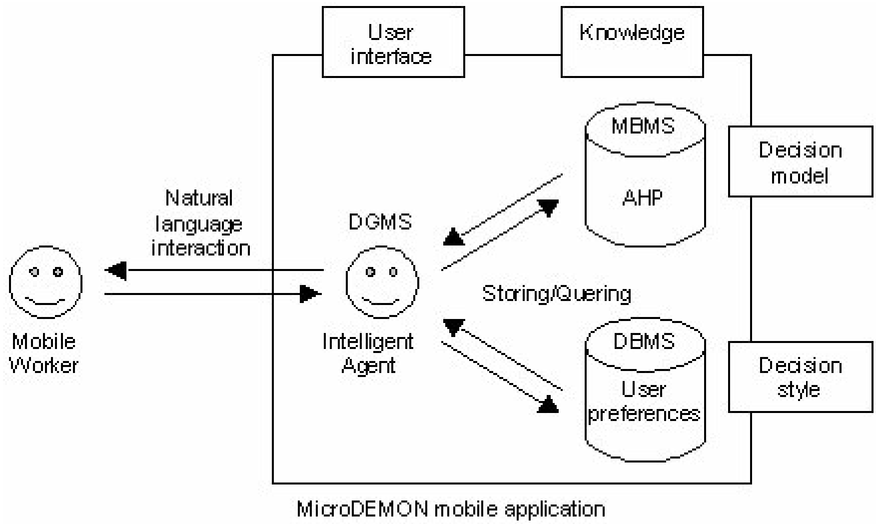


Figure 13.1. Architecture of the MicroDEMON

13.4.2 The MicroDEMON Decision Model

It has been rather convincingly demonstrated by numerous empirical studies that the human judgment is based on intuitive strategies as opposed to theoretically sound reasoning rules (Bell *et al.* 1988, Dawes 1988, Kahneman *et al.* 1982), but decision making can also be improved by framing rigorously the problem that must be faced.

Building a model of a decision problem allows for applying scientific knowledge that can be transferred across problems and often across domains. A decision-making model represents a measure of preferences over decision objectives, all available decision options, a measure of uncertainty over variables in the decision, and the outcomes.

Preference is widely viewed as the most important concept in decision-making. Outcomes of a decision process are not all equally attractive and it is crucial for a decision maker to examine them in terms of their desirability. Preferences can be ordinal (*e. g.* more income is preferred to less income), but even when they consist of just a single attribute but the choice is made under uncertainty, expressing preferences numerically allows for tradeoffs between desirability and risk. For what concerns decision options, often they can be enumerated (*e. g.* a list of possible suppliers), but sometimes they are continuous values of specified policy variables (*e. g.* the amount of raw material to be kept in stock). Listing the decision options is an important element of model structuring. About uncertainty, finally, it is one of the most inherent and most prevalent properties of knowledge, originating from incompleteness of information, imprecision, and model approximations made for the sake of simplicity. It would not be an exaggeration to state that real-world decisions not involving uncertainty belong to a truly limited class (Druzdel and Flynn 2000). The decision process rests on the assumption that a good decision is the one that results from a good decision-making process that considers all important factors and is explicit about decision alternatives, preferences, and uncertainty.

In everyday enterprise work, we often face the task of selecting an action to be performed from a set of feasible alternatives. The practical difficulty is that such decision problems are not easy to solve because they involve multiple objectives. Then, the decision-making model adopted by MicroDEMON is one of the most complete and manageable multifactorial models (Keeney and Raiffa 1976) to frame and solve both qualitative and quantitative decision problems: the analytic hierarchy process (AHP; Saaty 1990). The AHP decision model can be explained through the following steps:

1. problem is decomposed into a hierarchy of goals, alternatives, criteria and subcriteria;

2. data is collected from experience corresponding to the hierarchic structure, in a pairwise comparison of alternatives on a gradation scale, whose typically range is 1 to 9, for qualitative-to-quantitative comparison of alternatives;

3. the pair wise comparisons of various criteria generated at step 2 are organized into a square matrix, where the diagonal elements are equal to 1; the criterion in the i-th row is better than the criterion in the j-th column if the value of the element (i, j) is more than 1, otherwise the criterion in the j-th column is better than the criterion in the i-th row; the (j, i) element of the matrix is the reciprocal of the (i, j) element;

4. the principal eigenvalue of the matrix is computed and the corresponding right eigenvector of the comparison matrix gives the relative importance of the various compared criteria; the elements of the normalized eigen vector are termed weights with respect to the criteria or subcriteria and ratings with respect to alternatives;

5. the consistency of the matrix is then evaluated through an index that should be less that 0.1 and the reckoning of which is based on the maximum eigenvalue of the matrix itself; comparisons made by this method are subjective and AHP tolerates inconsistency through the amount of redundancy in the approach;

6. the ratings of each alternative are multiplied by the weights of the sub-criteria and aggregated to get local ratings with respect to each criterion; the local ratings are then multiplied by weights of the criteria and aggregated to get global ratings.

In other words, AHP produces weight values for each alternative based on the judged importance of one alternative over another with respect to a common criterion. This model provides an effective structure in which alternative decisions and the implications of taking those decisions can be laid down and numerically evaluated also if they are qualitative choices. It also helps both the user and the intelligent agent to form an accurate, balanced picture of the risks and rewards that can result from a particular choice. Another selling point of the AHP model is its intrinsic parallelism, which make it suitable for an implementation by the object-oriented programming (OOP) approach: model blocks can be managed and manipulated by the intelligent agent when interacting with the human decision-maker, which can then perform tasks by following an intuitive path to framing the decision problem instead of a too rigid and repetitive data inputing task, like in other intelligent DMSS (Mentzas 1997).

13.4.3 A Multiagent Java-based Distributed Active DMSS

Information sharing and knowledge management, enabled by new media tools, become more and more important in complex supply-chains management. They also may become of great importance for large networks spanning enterprises, research institutions and economic sectors.

The most remarkable trends in communication have been the huge popularity of the Internet and the growth of digital cellular telephony usage. There is a strong demand to combine these two in the form of mobile Internet access. Then, the focus of our distributed approach was not only on the development of a user-centred decision support system embedded in a mobile device, but also in extending communication from this strange team to similar groups within the same enterprise, providing spreaded group decision-making support by means of an XML-based modeling language for wireless information interchange.

Extensible Markup Language (XML) was designed to deliver structured, possibly complex content over the Web while still being easy to implement. It does not contain any functionality, but it is used as a data description, interchange, and storage format, since it provides a way to organize information, so that it can be automatically processed. The most significant advantages of XML can be summarized in the following three elements (Marttila and Vuorimaa 2000):

- flexibility, since XML can be used for many different purposes in various services, systems, and platforms providing also an interchange format between applications;
- reusability, since the document format can remain unchanged among a number of documents, while the content is changing;
- business intelligence (the semantics of information), since the processing of the information can be automated.

XML is, first of all, a device and platform independent method for describing information: the information coded in XML is always the same, but the presentation can be chosen according to the facilities of the device. In MicroDEMON, XML is used to represent information about decision problems tackled and to exchange this information with colleagues by means of the enterprise repository server (*i. e.* an XML-formatted document database implemented as a Java Server Page technology) that stores and categorizes all the decision problems the mobile workers deal with.

This new type of cooperative distributed user-centered working environment, highly context aware, adaptive and knowledge supported would lead mobile empowered people in companies to come together without any barriers to exploit business opportunities by solving decision-making problems. This cooperative environment is seen as seamlessly embedded in cooperative business processes of networked organizations. Thus, it is aimed to bridge the gap between:

- person-and-group-oriented cooperative support, and
- interorganizational business processes and business process management.

We tried to provide the life-cycle support of interrelated communication, interaction, judgment and decision processes in both cooperative work and business value systems: a single MicroDEMON application can not only work in a team with the mobile worker without any enterprise support, since the decision model provided and the intelligent assistant interface are evolved enough to make the worker able to deal with varying decision context and to adapt to changing information without lack of effectiveness, but it can also enable the function to be aware of the enterprise group of headquarters or mobile decision makers to share and to receive updated information to make more effective decision-making and in-the-field business performance, since group decision-making operationally means increasing the speed at which decisions are reached without reducing, and hopefully enhancing, the quality of resulting decisions (De Sanctis and Gallupe 1987).

The multiagent ADMSS system that araises from the interconnection of multiple MicroDEMONs (see Figure 13.2) permits connection and integration of all the enterprise levels of decision, from the operational (*e. g.* mobile workers) to the strategic one (*e. g.* company managers), through the tactical one (*e. g.* marketing area managers). This should result in an improvement of the company effectiveness in activities on the field. In this sense, Java technology is the most suitable platform to approach interoperability among devices (*i. e.* enterprise server, headquarter's PCs, mobile workers' devices, and so on).

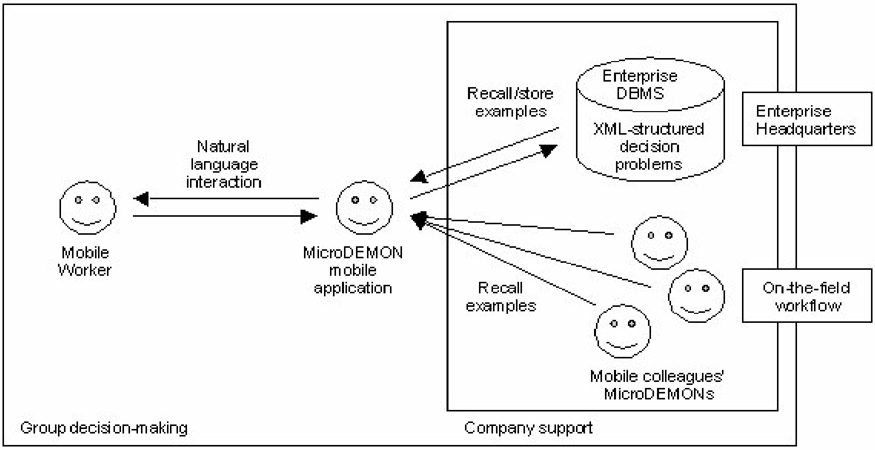


Figure 13.2. The multiagent ADMSS architecture made of connected MicroDEMONs

13.5 Contribution to the Intelligent DMSS Field

As human work takes place in crossorganizational workspaces, and knowledge and information sharing become critical, there is a need for integrating knowledge management, knowledge sharing and learning support in cooperative environments. In other words, business intelligence needs to be supported also in a mobile business environment.

From Active DMSS, a kind of proactive agent-based DMSSs, we moved a step forward in the evolution of Intelligent DMSSs (Hollnagel 1987; Buckner and Shah 1991; Gottinger and Weimann 1992): MicroDEMON is an ADMSS that is equipped with natural language-processing skills to make more natural and then effective the agent support during interaction with the decision maker when performing problem solving (see Appendix A). Moreover, by extending the concept of ADMSS to embrace a multiagent system perspective, we also connected MicroDEMONs to give raise to a multiagent ADMSS that makes organizations able both to manage different decision levels of a mobile business process, from the mobile worker to the headquarters managers, and to support group decision making at all stages.

This distributed ADMSS approach, which is based on an adequate architecture, modular components and intelligent agents-based applications, is revolutionary in the sense that it is user-centered instead of enterprise-centered. Due to great instability in the mobile network connection and variability in on-the-field decision environments mobile business processes can be properly managed only by moving central decision-making activity to mobile workers without losing the strenght of group decision making, since it is simulated by the virtual interaction between the user and the agent embedded in the mobile device. Enabling of enterprise decision-making integration in distributed work contexts is a very important feature our system can show by means of a Web-based infrastructure supporting interoperability among mobile workers and headquarters decision-makers.

Embedding an ADMSS architecture in a hardware/software resource limited mobile device without losing information analysis and decision-making complexity was another important issue we faced successfully and this opened new kind of portable, mobile DMSS.

13.6 Conclusions

The recent massive use of wireless technology in the business domain strongly modified organization and management of work (Perry *et al.* 2001), making it critical both to gather decision-problem data and to share them among human business agents in real-time (Pica and Kakihara 2003). To help mobile decision-makers, we developed MicroDEMON, an application for mobile business that implements the ADMSS architecture outlined in Pistolesi (2002) and can be installed on any mobile device equipped with a Java Virtual Machine. By a natural language interaction the human mobile worker can manage the complexity of the multicriteria decision model adopted to solve the problem, namely, AHP (Saaty 1990). With respect to most part of the few current DMSS for mobile business, which are company-centered and not truly distributed (San Pedro *et al.* 2003), MicroDEMON is a user-centered application that enables group decision-making at different levels, allowing global enterprise decision complexity to be managed with mobile technology.

In this chapter we provided an overview of ADMSSs and decision making in mobile business, then we described the MicroDEMON decision support system to show how useful and promising is the adoption of natural language-processing agents as the building blocks of ADMSSs for decision-making support.

Appendix A

Below is an example session between the mobile worker and MicroDEMON to solve a simple structured and operational decision problem.

MDEMON: Hi guy! Please, try to describe the decision problem we have to deal with.
USER: Choose the best product fitting the customer's requests.
MDEMON: I checked the internet repository but I didn't find anything useful. Then, which task do you wish to perform now?
USER: Insert criteria.
MDEMON: You must insert criteria, one at a time, to evaluate several alternatives; insert a criterion and press OK, then you could insert more criteria.
USER: Usage complexity.
MDEMON: You insert 'Usage Complexity' as a criterion. Any more?
USER: Work improvement.
MDEMON: We have 'Usage Complexity' and 'Work improvement'. Again?
USER: No.

MDEMON:	Which task do you wish to perform now?
USER:	Insert alternatives.
MDEMON:	Please, insert an alternative.
USER:	Software #1.
MDEMON:	Ok, we have ‘Software #1’. Any more?
USER:	Software #2.
MDEMON:	Very well. ‘Software #1’ and ‘Software #2’. Do you wish to insert any more alternatives?
USER:	No.
MDEMON:	Which task do you wish to perform now?
USER:	Insert criteria.
MDEMON:	You must insert criteria, one at a time, to evaluate several alternatives; insert a criterion and press OK, then you could insert more criteria. Please, remember you inserted ‘Usage complexity’ and ‘Work improvement’ criteria, till now.
USER:	Price.
MDEMON:	Ok, criteria are ‘Usage Complexity’, ‘Work improvement’ and ‘Price’. Any more?
USER:	No.
MDEMON:	Which task do you wish to perform now?
USER:	Build or modify the criteria hierarchy.
MDEMON:	The following criteria (Usage complexity, Work improvement, Price) have been inserted till now. Do you wish to gather some under the same father criterion?
USER:	Yes.
MDEMON:	Please, choose a son criterion.
USER:	Work improvement.
MDEMON:	The father criterion is in the same list or add it?
USER:	Add it.
MDEMON:	Very well. Insert the father criterion.
USER:	Software benefits.
MDEMON:	The new criterion ‘Software benefits’ is the father of ‘Work improvement’. Right?
USER:	Yes.
MDEMON:	More criteria to gather under ‘Software benefits’?
USER:	Yes.
MDEMON:	Choose another son criterion for ‘Software benefits’.
USER:	Price.
MDEMON:	We have ‘Work improvement’ and ‘Price’ as sons of ‘Software benefits’. Right?
USER:	Yes.
MDEMON:	Any more gathering?
USER:	No.
MDEMON:	Which task do you wish to perform now?
USER:	Evaluate criteria.
MDEMON:	Very well. Which kind of scale (Adjectives, Gradations, Numbers) do you wish to apply to couple comparisons?
USER:	Gradations.
MDEMON:	Ok. Now select a criterion to compare.
USER:	Usage complexity
MDEMON:	Very well. Due to the hierarchy you build, you can compare ‘Usage complexity’ only with ‘Software benefits’. Do you wish to change the criteria hierarchy?

USER:	No.
MDEMON:	Ok. Tell me which is the more important criterion between ‘Usage complexity’ and ‘Software benefits’.
USER:	Software benefits.
MDEMON:	Now, tell me how much important is ‘Software benefits’ than ‘Usage complexity’.
USER:	4 (on a 9-level graduated bar)
MDEMON:	(Stores a value of 4 for the couple SB-UC and a value of 1/4 for UC-SB) Another criterion to evaluate?
USER:	Yes.
MDEMON:	Ok. At this stage, you can only compare ‘Work improvement’ with ‘Price’. Should I proceed?
USER:	Yes.
MDEMON:	Ok. Tell me which is the more important criterion between ‘Work improvement’ and ‘Price’.
USER:	Price.
MDEMON:	How much important is ‘Price’ than ‘Work improvement’?
USER:	8 (on a 9-level graduated bar)
MDEMON:	(Stores a value of 8 for the couple P-WI and a value of 1/8 for WI-P) Evaluation of criteria is completed. Do you agree?
USER:	Yes.
MDEMON:	Which task do you wish to perform now?
USER:	Evaluate alternatives.
MDEMON:	Very well. We’ll use the Gradations scale you already choose for criteria to compare alternatives. Do you agree?
USER:	Not sure.
MDEMON:	Remember you can change the decision process at any step. Then you can change the evaluation scale but, please, take into account that then you must compare again criteria you already evaluated. Do you wish to change the scale?
USER:	No.
MDEMON:	Ok. Which is the best alternative between ‘Software #1’ and ‘Software #2’ with respect to the criterion ‘Price’?
USER:	Software #1.
MDEMON:	Ok. How much is ‘Software #1’ the best with respect to ‘Software #2’ for the criterion ‘Price’?
USER:	6 (on a 9-level graduated bar)
MDEMON:	(Stores a value of 6 for the couple S1-S2 and a value of 1/6 for S2-S1 for the criterion P) Evaluate again?
USER:	Yes.
MDEMON:	Ok. Now, tell me which is the best alternative between ‘Software #1’ and ‘Software #2’ with respect to the criterion ‘Usage complexity’.
USER:	Software #2.
MDEMON:	Ok. How much is ‘Software #2’ the best with respect to ‘Software #1’ for the criterion ‘Usage complexity’?
USER:	3 (on a 9-level graduated bar)
MDEMON:	(Stores a value of 1/3 for the couple S1-S2 and a value of 3 for S2-S1 for the criterion UC) Another evaluation?
USER:	Yes.
MDEMON:	Very well. Tell me which is the best alternative between ‘Software #1’ and ‘Software #2’ with respect to the criterion ‘Work Improvement’.
USER:	Software #2.

MDEMON:	Ok. How much is 'Software #2' the best with respect to 'Software #1' for the criterion 'Work Improvement'?
USER:	4 (on a 9-level graduated bar)
MDEMON:	(Stores a value of 1/4 for the couple S1-S2 and a value of 4 for S2-S1 for the criterion WI) Evaluation of alternatives has been completed, since you can compare alternatives with respect to son criteria only. At this stage, we have all the information needed to find the best decision. Do you wish to proceed with the solution reckoning or to change something?
USER:	Proceed with the solution reckoning.
MDEMON:	The best solution is 'Software #1' with a value of 0,778, while the 'Software #2' alternative took a 0,222 value. Do you wish to see a graph?
USER:	Yes.
MDEMON:	(Shows a bar diagram about the solution reckoned)
USER:	Ok.
MDEMON:	(Sends the decision problem data to the internet repository at the enterprise headquarter) I sent the problem data to the internet repository for our colleagues. Do you wish to change some data of this problem and simulate decision again?
USER:	No, thanks.
MDEMON:	Another problem to face or quit?
USER:	Quit.
MDEMON:	Bye.

References

Alloway RM, Quillard JA (1983) User manager's systems needs. Management Information Science Quarterly 6:27-41.

Angehrn AA (1993) Computers that criticize you: stimulus-based decision support systems. Interfaces 23:3-16.

Anthony RN (1965) Planning and control systems: a framework for analysis. Harvard University Press, Cambridge (MA).

Barnard PJ, Hammond NV, Morton J, Long BJ, Clark IA (1981) Consistency and compatibility in human-computer dialogue. Int Journal of Man-Machine Studies 15:87-134.

Bell D, Raiffa H, Tversky A (eds.) (1988) Decision-making. Cambridge University Press, New York.

Bobrow D (1968) Natural language input for a computer problem-solving system. In: Minsky M (ed.) Semantic information processing. MIT Press, Cambridge (MA.).

Buckner G, Shah V (1991) Intelligent decision support systems. J Comput Inf Syst 31:61-66.

Carroll JM, Campbell RL (1989) Artifacts as psychological theories: the case of human-computer interaction. Behavioral Information Technology 8:247-256.

Chin DN (1998) Intelligent interfaces as agents. In: Maybury MT, Wahlster W (eds.) Readings in intelligent user interfaces. Morgan Kaufmann, San Francisco (Ca.).

Cowan DA (1991) The effect of decision-making styles and contextual experience on executives' descriptions of organizational problem formulation. Journal of Management Studies 28:463-483.

Daft RL, Lengel RH (1986) Organizational information requirement, media richness and structural design. Management Science 32:554-571.
Davis GB, Olson MH (1985) Management information systems. McGraw-Hill, New York.
Dawes RM (1988) Rational choice in an uncertain world. Hartcourt Brace Jovanovich, San Diego (CA).
De Sanctis G, Gallupe RB (1987) A foundation for the study of group decision support systems. Management Science 33:589-606.
Druzdel MJ, Flynn RR (2000) Decision support systems. In: Kent A (ed.) Encyclopedia of library and information science. Marcel Dekker, New York.
Edwards JS, Duan Y, Robins PC (2000) An analysis of expert systems for business decision-making at different levels and in different roles. European Journal of Information Systems 9:36-46.
Elliott C, Brzezinski J (1998) Autonomous agents as synthetic characters. AI Mag 2:13-30.
Er MC (1988) Decision support systems: a summary, problems and future trends. Decision Support Systems 4:355-363.
Gorry GA, Scott-Morton MS (1971) A framework for management information systems. Sloan Management Review 13:55–70.
Gottinger HW, Weimann P (1992) Intelligent decision support systems. Decision Support Systems 8:317-332.
Hayes J, Allinson CW (1994) Cognitive style and its relevance for management practice. Br J Management 5:53-71.
Hollnagel E (1987) Information and reasoning in intelligent decision support systems, Int Journal of Man-Machine Studies 27:665-578.
Kahneman D, Slovic P, Tversky A (eds.) (1982) Judgment under uncertainty: heuristics and biases. Cambridge University Press, New York.
Keen PGW (1987) Decision support systems: the next decade. Decision Support Systems 3:253-265.
Keeney RL, Raiffa H (1976) Decisions with multiple objectives, preferences and value tradeoffs. Wiley, New York.
Maes P (1994) Modeling adaptive autonomous agents. Journal of Artificial Life 1.
Maes P (1998) Agents that reduce work and information overload. In: Maybury MT, Wahlster W (eds.) Readings in intelligent user interfaces. Morgan Kaufmann, San Francisco (CA), pp 525-535.
Marttila O, Vuorimaa P (2000) XML based mobile services. In: The 8th International Conference in Central Europe Computer Graphics, Visualization, and Interactive Digital Media.
Mentzas GN (1997) Intelligent process support for corporate decision-making. Journal of Decision Systems 6:117-138.
Mintzberg H, Raisinghaniand D, Theoret A (1976) The structure of unstructured decision processes. Administrative Science Quarterly 21:246-247.
Norman DA (1994) How might people interact with agents. Communications of the ACM 37:68-71.
Norman DA (1998) The invisible computer. MIT Press, Cambridge (MA).
Nutt PC (1990) Strategic Decisions made by top executives and middle managers with data and process dominant styles. Journal of Management Studies 27:173-194.

Perry M, O'Hara K, Sellen A, Brown B, Harper R (2001) Dealing with mobility: understanding access anytime, anywhere. ACM Trans on Computer-Human Interaction 8:323-347.

Pica D, Kakihara M (2003) The Duality of mobility: understanding fluid organizations and stable interaction. In: ECIS 2003, Naples, Italy.

Pistolesi G (2002) How synthetic characters can help decision-making. In: Mora M, Forgionne G, Gupta J (eds.) Decision-making support systems: achievements, trends and challenges for the new decade. Idea Group, Hershey (PA), pp 239-256

Preece J (1994) Human-computer interaction. Addison-Wesley, Reading (MA).

Raghavan SA (1991) JANUS, a paradigm for active decision support. Decision Support Systems 7:379-395.

Rice R (1992) Task analyzability, use of new media, and effectiveness: a multi-site exploration of media richness. Organizational Science 3:475-500.

Saaty TL (1990) How to make a decision: the Analytic Hierarchy Process. European Journal of Operational Resesearch 48:9-26.

Sage AP (1991) Decision support systems engineering. John Wiley & Sons, New York.

San Pedro J, Burstein F, Zaslavsky A (2003) Support for real-time decision-making in mobile business applications. In: Proceedings of the Second International Conference on Mobile Business, Vienna, Austria, June 23-24, pp 383-392.

Sasse MA (1992) Users' models of computer systems. In: Rogers Y, Rutherford A (eds.) Models in the Mind. Academic Press, London.

Scheer A (1998) Business process engineering: reference models for industrial enterprises. Springer, Tokyo.

Schneiderman B (1988) Direct manipulation: a step beyond programming languages. IEEE Trans on Computing 16:57-69.

Simon HA (1960) The new science of management decision. Harper & Row, New York.

Sprague RH (1980) A framework for the development of decision support systems. Management Information Systems Quarterly 4:7-32.

Turban E, Aronson JE (1998) Decision support systems and intelligent systems. Prentice Hall, Upper Saddle River (N.J.).

Tyrvainen P, Veijalainen J (2003) Human constraints for mobile communications. In: Proceedings of the 14th ITS Regional Conference, Helsinki, August 23-24.

Vassileva J (1996) A task-centered approach for user modeling in a hypermedia office documentation system. User Modeling and User-Adapted Interaction 6:185-223.

Weizenbaum J (1976) Computer power and human reason. Freeman, San Francisco (CA.).

14

Using System Dynamics and Case-based Reasoning (CBR) to Build an Intelligent Decision-making Support System (i-DMSS) that Improves Strategic Public Decisions

Carlos A. Legna[1] and Carina S. González[2]

[1] Department of Institutional Economics, University of La Laguna, Spain. 38071, La Laguna, Spain. Phones: +34 922 317334 FAX +34 922 317427. Email: clegna@ull.es
[2] School of Computer Science, University of La Laguna, Spain, 38071, La Laguna, Spain. Phones: +34 922318234 FAX +34 922318228.Email: cjgonza@ull.es

This chapter presents an intelligent decision support system (IDSS) that helps decision makers to identify key issues and to improve policy-making processes, particularly with respect to strategic decisions. The IDSS combines artificial intelligence (AI) techniques with qualitative models and system dynamics simulation, and allows quantitative and qualitative variables to be integrated into a comprehensive methodology, necessary to design strategies and policies. The results of this study are currently in the process of being implemented.

14.1 Introduction

This chapter is part of a research project that focuses on the application of knowledge management (KM) to improve the decision-making processes related to strategic public decisions (SPD) that take place at the highest levels in regional and local government. These decisions require the management of qualitative and quantitative information, the use of complex systems and, in most cases, the solution of nonprogrammed problems (Simon 1977). In this chapter we describe the IDSS that we have built, focusing on these characteristics of strategic decisions; we also explain how it works and produces results in each phase of the systematic decision-making process. Its first steps were proposed by Simon (1977) but completed later, as his earlier work did not take into account implementation and monitoring. Adding these two phases to the ones proposed by Simon, the process includes the following phases: 1) Intelligence; 2) Design; 3) Choice; and 4) Implementation. Following Turban *et al.* (2002, p.49), we consider the monitoring as the first phase applied to the last one. It is important to emphasize that all the phases and subphases of the process are iterative.

The application of Simon's systematic decision making in our case is probably more complex than when it is used for organizations, because we are studying entire societies where social, political, environmental and economic variables, to name but a few, are involved. A description of how the IDSS works and produces outputs in each phase (except with respect to implementation) is included in the following sections. The last one explains how the IDSS improves the decision-making process, taking into account how human beings make decisions.

14.2 Intelligence Phase

The goal of the intelligence phase is to produce a problem statement, which will then be used as the input for the next phase (design). Our approach produces this output by dividing the intelligence phase into the following subphases: 1) Selecting the objectives; 2) Understanding the regional structure (integrating social, economic, political and ecological variables, among others); 3) Predicting the possible futures of the regional social system by creating different scenarios; and 4) Identification of the problem(s) to solve.

14.2.1 Selecting the Main Objectives

When the systematic decision process is applied to organizations, the first subphase is the identification of organizational objectives. In our case, societal objectives must be determined, which requires decisions at a high political level. This required several discussions and brainstorming sessions with the authorities to select primary long-term social objectives. For instance, in a study done for the Canary Islands, Spain, (Legna and Rivero-Ceballos 2001) to help prepare its negotiating strategy in the European Union (EU), "*sustainable development of the population's quality of life*" was selected as the Canarian government's main objective. Evidently, this is a broad goal. The next step consisted of translating this main objective into operational ones. To do so, it was necessary to answer the following question: at present, what are the most important components of the quality of life for the Canarian population? The answer was expressed as a vector that contains the following variables: V5=**gross domestic product** (GDP); V15=**employment and wage levels**; V9=(-)**costs derived from insularity**, double insularity[1] and the distance from the European continent; V25=(-)**mortality rate**; V32=**human capital**; V40=**female rate of activity**; V41=**general cultural level**; V43=**urban, rural and marine environment** (including beaches). The sustainable increase of the values or indicators of each of these variables was defined as the operational objectives[2].

---

1 "Double insularity" refers to the fact that the Canary Islands is an archipelago that produces transport and communications costs in addition to the ones derived from the segmentation of the markets (which, for example, make it difficult to take advantage of scale economies).

2 Note that the values of the variables V9 and V25 are preceded by a minus sign. "V" indicates "Variable" and the numbers are assigned in the qualitative model that will be explained in 14.2.2.1

Briefly explained, in order to select these operational objectives, the following steps were implemented and integrated: a) scientific studies, mainly diagnostics of the society, were carried out to detect variable "candidates" to be included in the vector; and b) value judgments were applied in order to make a final selection. For instance, the first step detected that the **unemployment rate** in the Canary Islands was higher than that of Spain and the **real wages** were lower. In order for these variables to be included in the vector, in other words, accepted as operational objectives, the regional authorities had to consider the state of these variables to be negative. To select the final vector of problems, discussions were held that included the participation of the technical team and the decision makers. Step a) does not always precede step b); sometimes variables/objectives are detected during step b), the political level, leading to their study on the scientific level, step a).

The following section explains how our approach detected strategic problems on the scientific level, integrating qualitative and quantitative information with the knowledge and values of the political leaders.

14.2.2 Understanding the Regional Structure: Integrating Social, Economic, Political and Ecological Variables

After establishing the operational objectives, the next step is to gain insight into the regional structure and the key feedback loops between variables that affect accomplishing those objectives. This is done by building "qualitative models" that integrate qualitative and quantitative information and then applying system dynamics to them.

14.2.2.1 Qualitative Models

The following is a brief description of how these models are built.

First, research is done to detect: a) the variables that produce changes in the operational objectives; and, b) the relationships between all the variables (including the ones that are built-in to the objectives vector). The latter refers to building the functions of the model.

Relationships between variables are established such as the following: V3=f(V1;V4;V8;-V14). This means, for instance, that a change in the variable V1 affects or produces a change in V3. The minus sign in V14 means that a change in this variable produces a negative change in V3. When they express behaviors, the relationships between the variables have to be understood as the "*usual value or course of action in a given situation (…) John usually takes his car if the weather is nasty*" (Yager 1986). For example, "traditional farmers often use technology that produces low yield and a degradation of the lands as well as an inefficient use of water resources". Then systems dynamics are applied to the model to identify the key strategic loops in the regional structure.[3]

Instead of discarding quantitative information and econometric models, we integrate their results into the qualitative ones. For instance, to build the function of

[3] The description of the application of system dynamics to the model is based on Kljajic, Legna and Skraba (2002 and 2003).

the variable 19=**value of the production of the construction sector**, we use an econometric model created for the Canary Islands. Strategic decisions require working with qualitative information that is frequently crucial. Our approach is designed to overcome the disparity of decision-making processes that use very refined quantitative models to treat the quantitative information and raw instruments to understand the qualitative aspects of the issue. In this way it is possible to have a comprehensive view of the main variables and issues that involve the strategic decision process.

14.2.2.2 Applying System Dynamics to Qualitative Models

Applying system dynamics to the main variables of the Canary Islands case allows us to identify the main relationships and feedback loops between them (Figure 14.1). There are six main loops, three reinforcing and three balancing. The positive loop which interconnects **GDP**, **development policy** and **tourist market** includes the forces that drive the economic cycle, which up until the present has had a tendency to grow. The second positive feedback loop between **GDP**, **development policy** and **agriculture and industry production** leads to enhanced **industrial and agricultural production**. There is also a third positive feedback loop, which includes the **tourist market**, **employment opportunities and quality of life**, **immigration**, **population**, **human resources** and **agriculture and industry**. This reinforcing loop is an interaction between **population**, **employment opportunities** and **immigration**. The model is balanced by the negative feedback loops. The first one includes the **tourist market**, **preservation of natural resources** and **regional attractiveness**. As the system tends to increase the **tourist market** in the islands, the process is moderated by **environmental attractiveness**, which is diminished by the overcrowding of the region. A similar negative effect is the loop that interconnects **agricultural production** and **preservation of natural resources**. There is also an important negative feedback loop between the **GDP**, **population** and **preservation of natural resources**. Therefore, the model highlights the main variables and loops that determine the dynamic of the Canary archipelago; and it also underlines the importance of properly balancing these loops to attain a sustainable development of the region[4]. Unfortunately, at present, the relationships between these forces are leading the system to an undesirable scenario, and so the development model needs to be changed.

These are the first conclusions at an aggregate level. The next step is to build submodels in order to obtain more detailed results. In the Canary Islands study the following submodels were constructed: **Population**, **Tourism**, **Agriculture**, **Environment** and **GDP**.

[4] A more detailed examination of the variables was presented in our previous work (Kljajić *et al.* 2002).

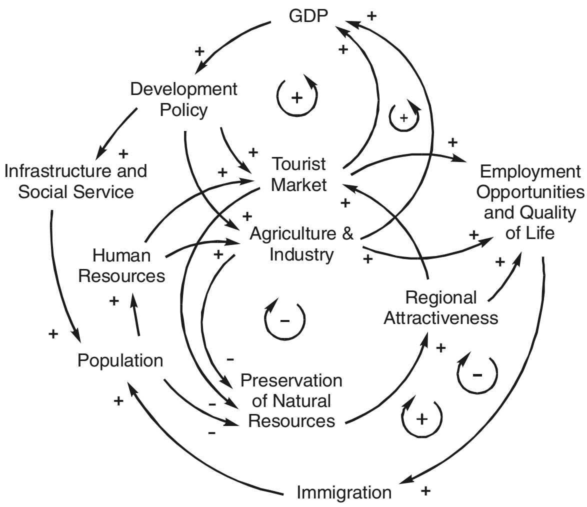


Figure 14.1. usal loop diagram of the Canary Islands

The following is a brief description of the population submodel. The inhabitants are divided into three categories: population that is under 16 years old, active population between 16 and 65 and the elderly, over 65. The total number of residents in the islands is approximately 1.8 million. The main variables that characterize the dynamics of the population are the birth rate coefficient, mortality coefficient and immigration. The Canary Islands acreage is 7447 km^2. The demographic density is 240 people/km^2. The demographic density of Spain, for example, is 80 people/km^2. If we include the “dynamic” population of about 10 million tourists that visit Canary Islands every year we can see that the demographic density is relatively high. In addition to the high demographic density, it must be mentioned that the population is relatively young in comparison with other Spanish regions.

14.2.3 Predicting the Possible Futures of the Regional Social System by Creating Different Scenarios

The previous steps allow key variables and loops to be identified from the perspective of the desired future for the Canary Islands. The subsequent steps build scenarios doing a “what if” study of the effects of possible paths of the crucial variables. The following describes how the scenarios were built, using the population submodel. We developed two simple scenarios regarding possible population trends:

- Population growth remains constant during the following 25 years. This is likely to occur if there are no other external forces to influence the birth and mortality coefficients. Expected exponential growth should have a negative impact on the environment resulting from population overcrowding.

- Population growth is controlled and moderately reduced in order to achieve stable population numbers. This should help to preserve the island from population overcrowding that negatively impacts the environment.

Figure 14.2 represents these two possible population growth scenarios. The curve marked by 1 represents the scenario where the birth coefficient remains the same for the next 25 years. As we can see, this causes exponential growth. By increasing the number of inhabitants, sustainable development is much harder to achieve. The quality of life under this scenario will also decrease. In our second scenario, marked by 2, the birth rate moderately decreases during the next 25 years and the total population finally stabilizes. The growth in the second scenario is more moderate.

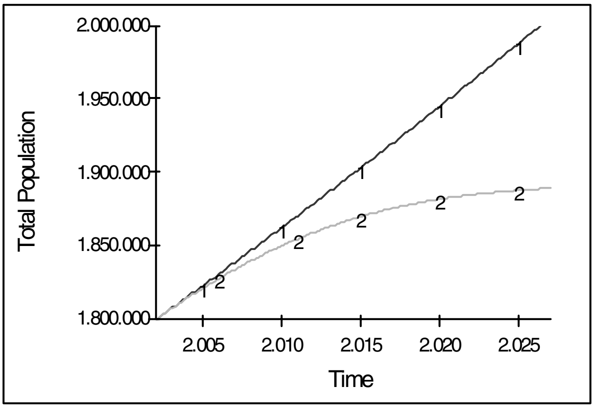


Figure 14.2. Total population 1 ~ Constant birth coefficient, 2 ~ decreasing birth coefficient

According to the projections of the first scenario, the “working age” population will be stable during the following 20-25 years (if working-age immigration is not elevated); the weight of the elderly will increase while the weight of the youth will decrease. If so, the supply of labor, in quantity, is not substantially affected by the increase of the population. This conclusion is stronger in the case of the other scenarios that were built. The construction of the scenarios highlighted that the increase in the GDP will necessarily need an increase in labor productivity. This need is more acute in the second scenario. The change of the population pyramid structure towards an older one affects the demand for public and private services. The quality and the quantity of public services, such as education, health and urban services, need to be adapted to the new structure of the population. That is, the public sector ought to design and implement strategic plans in these sectors by taking into account the new conditions. The same comments apply to some private sectors, like tourism, personal services (paramedical, cultural services, house keepers, and so on). The interesting point is that these new conditions are at once a possibility, an opportunity, and a “window” open to the development of different and more qualified jobs in the labor market.

14.2.4 Identifying the Problem (or Problems) to Solve

To this point, the IDSS has allowed us to select the main goal and the operational objectives; to understand the regional structure and the relationships between the key variables that affect accomplishing the objectives; and to build scenarios of the possible futures of the social system. Based upon these results, the next step is to identify the problems to solve, in order to progress toward accomplishing the operational objectives. We will explain how to accomplish this using the example of the population scenarios discussed in the previous section. These scenarios allow us to detect some questions (or problems) that the decision makers need to resolve.

They are the following (among others):

- How to produce health, education and urban services for a more aged population?
- How to generate qualified employment by producing goods and services for the elderly population
- How to reduce the pressure on the territory if the population continues to increase?

By taking into account the population submodel and its scenarios, it was possible to detect 3 strategic problems (although they are not the only ones). The same approach can be used to detect others, which have to be added to the list. In turn, these problems may be disaggregated into subproblems. These results are the inputs for the next phase (design).

14.3 The Design Phase

The design phase involves finding or developing and analyzing possible responses to the problems identified in the previous phase. The bibliography on DSS frequently mentions that this includes understanding the problem and testing the feasibility of the solutions. The problem is that most strategic decision issues are totally unstructured and cannot be tested. The strategic problems that decision makers usually have to confront are similar to the one that the President John F. Kennedy had to solve during the Cuban Missile Crisis.

After the disastrous "Bay of Pigs" invasion, the Soviet Union decided to take advantage of the situation to install missiles in Cuba. The USA was slow to discover this operation and President Kennedy ordered a blockade of Cuba. Before making this final decision various alternatives were studied (Bernoux 1985, p.284): a) do nothing; b) diplomatic actions; c) to negotiate with President Castro; d) to negotiate with the USSR the removal of Soviet bases in Cuba in exchange for removing American bases in Italy and Turkey; e) a ”surgical” air strike on Cuba; f) a naval blockade. Each choice would have produced different effects on government's goals but it would not have been possible to compare and quantify them. In addition, each course of action would have different costs and the information necessary to make the decision was incomplete and erroneous. As Allison demonstrated, in these conditions, it was impossible to come to a solution following a substantive rational process, as *homo economicus* (Allison 1988).

In this situation, how would it be possible to test a course of action without at least starting its implementation? How would it be possible to test the feasibility of an air strike on Cuba? When it is possible to conduct experiments in a laboratory, most aspects of a technical problem may be tested. But this is not possible for social and political systems. How can a totally new problem be tested? How would it be possible for President Kennedy to test the feasibility of a surgical air strike on Cuba, for instance? There is a large degree of uncertainty about the results of solutions to unfamiliar problems. Our approach tries to reduce the uncertainty in making these strategic public decisions, as well as reduce some of the biases that affect the decision-making process, as will be explained in the following sections.

14.3.1 Finding Solutions

The Case-based reasoning methodology may be a useful tool to improve the results of the decision-making process in these strategic situations, as will be explained in the following section.

14.3.1.1 Applying CBR

Decision support systems (DSS) are interactive computer-based information systems designed to help human decision makers. These systems process data and models in order to identify, structure, and solve semistructured or unstructured problems and make choices among different alternatives (Zolghadri *et al.* 2002).

With these types of applications, experts must evaluate and make decisions with the data provided by analysis tools. One way to create a useful tool is to represent the reasoning process in the form of rules and build an expert-knowledge based system. But knowledge based systems have several problems related to the extraction and representation of expert knowledge. Therefore, these systems are generally slow and usually cannot access huge amounts of information. *This is why we propose a case based reasoning method that resolves new problems by adapting past solutions to solve similar problems in the future* (Riesbeck and Schank 1989).

One advantage of this technique is that it does not require explicit knowledge of the domain. The extraction process is reduced to collecting historic cases and identifying relevant attributes of each case. First, we start with a small number of cases and then eliminate cases that are not useful, finally we add new ones. In addition, we can give explanations about why selections and decisions were made and use database techniques to administrate a large amount of information. Most importantly, the system learns by acquiring new knowledge from the cases. These features make the system easy to maintain and reuse.

14.3.1.2 Knowledge Representation

The information stored for each case is related to:

- The conditions that define the scenario;
- The problems that emerge from the particular conditions of the scenario;
- The descriptions of the solutions found for the problems and the decisions made;
- A result describing the state of the system after applying the suggested actions.

The case to be identified is represented by a collection of features, although some of the descriptions of these features may be. The expert knowledge and the different decision-making experiences in general are implicitly registered in the system in order to find the best solutions. Initially, we define cases for proposed scenarios, but the system is not limited to these propositions because it can learn and add new scenarios to the database as well as new solutions thought of by the user.

Each case has a class label, data particular to the case and general data that is shared by past cases. The class label is defined by the name of scenario. The total number of classes is equal to the total number of implemented scenarios. For example, in the population scenario the class label is “population”, its particular data (the data unique to this case) are the vector components of Ae $\{a_1 \ldots a_n\}$ and the general data it incorporates (data that is also found in past cases) are the components of the vector CI {indexes, relationships, rate of use, and rate of success}.

The structure of the case base is as follows:

Ce: Scenario class
Ae: Vector of the attributes that define the scenario $\{a_1 \ldots a_n\}$
Pe: Vector of problems detected in the simulation of the scenario $\{pe_1 \ldots pe_n\}$
Spe: List of proposed solutions $\{spe_1 \ldots spe_n\}$
Ep: Associated explanations
CI: Vector with additional case information {indexes, relationships, rate of use, rate of success}

The reasoning machine works as follows: the CBR system receives the current scenario’s data through the user interface and from the results of the simulation module. The user specifies the strategic issues to be analyzed and defines the characteristics of the scenario.

The CBR Cycle works as follows (Aamodt and Plaza 1994):

1. Similar case *retrieval* (a new problem is matched with similar cases stored in the case base. For example, how regions similar to the Canary Island have reduced their unemployment rate)
2. *Reuse* solutions proposed in past cases to try to solve the new problem.
3. *Revise* proposed solution (when necessary)
4. *Retain* the new solution as part of a new case

The cycle is completed by user intervention.

Each recovery case is assigned an index based on similarity. The selection algorithm used in our application is *the nearest neighbor***.** The similarity algorithm used by our selection engine has the following equation:

$$\frac{\sum_{i=1}^{n} w_i \times sim(f_i^I, f_i^R)}{\sum_{i=1}^{n} w_i}$$

Where,

> **w** = importance of the weight of an attribute
> **sim** = function of similarity of the attributes
> **f** = values for the attribute *i* in the old case and in the new one

The similarity metric uses multistep filtering. First of all, the classes that do not match the current scenario are eliminated from the searching space. Then the cases that have attributes most similar to the current case are selected, taking into account their significance to the scenario (weight of attribute). Afterwards, cases with problems similar to those of the current case are extracted from the set obtained in the previous step, leaving the CBR system with a reduced number of cases whose problems match that of the new case. This process can be described in the following way:

- Obtain a vector for the current scenario V0={Ce, Ae1;.... ;Ce, AeK}, with the relationship between the total number of similar attributes Ae and the total number of given attributes.
- Apply the second filter to each of the elements of the vector V0, and obtain a vector Vi, analogous to V0, with the relationship between the most similar problems that were found and the actual detected problems.
- Order the V0,Vi relationship in descending order in a result list, Rsim.

This algorithm recovers the most similar cases, where the most similar is the first element of Rsim. The case indexes used to find associated solutions are made from this vector. Once we have recovered the similar cases, the solutions associated to them must be adapted to the current case. The user must decide if these solutions are correct or not. This information is registered with other additional data that shows the usefulness of the case. If the system does not find any similar cases, the current case is stored as a new case. Similarly, if the proposed solutions do not satisfy the user new ones can be added through the user interface. This process closes the CBR cycle.

14.3.2 How the IDSS Integrates Qualitative Modeling, Simulation and AI Techniques

Our system is based on a new approach that includes both simulation and intelligent analysis techniques. It helps in the phases of the decision-making process as follows:

- **Intelligence phase** (*Problem Definition*, *Modeling* and *Simulation Module*): this is the first stage, where experts focus on the elaboration of qualitative models, the application of system dynamics and the preparation of the scenarios. Then the results of these tools are used to create simulations, following these steps: first, the user specifies the conditions of the scenario that will be simulated; second, the simulation module identifies the problems or "strategic questions"; then, the results of the simulations are stored in a database. They are the facts that define the actual case. In turn, this case will be processed by the Inference Engine module.
- **The design phase (***Knowledge Base Module* and *Inference Engine Module*)**:** the inference engine module, formed by representations of heuristic methods, follows this reasoning process: first it identifies the case, next it selects the case by similarity using inductive techniques, and finally it performs predictions.
- **Choice phase** (*User Interface*): the solutions are shown to the decision maker with explanations of the consequences of applying the different solutions. If the results and the explanations do not satisfy the user, it is possible to introduce new solutions. Thus, the system can be validated and can learn new solutions from the user.
- **Monitoring** (*User Interface*, *Knowledge Base Module*)**:** the results of the solutions are introduced in the CBR process, and the intelligence phase is repetead, in a continuous process. This is how the IDSS learns.

Interrelations among modules can be observed in Figures 14.3 and 14.4.

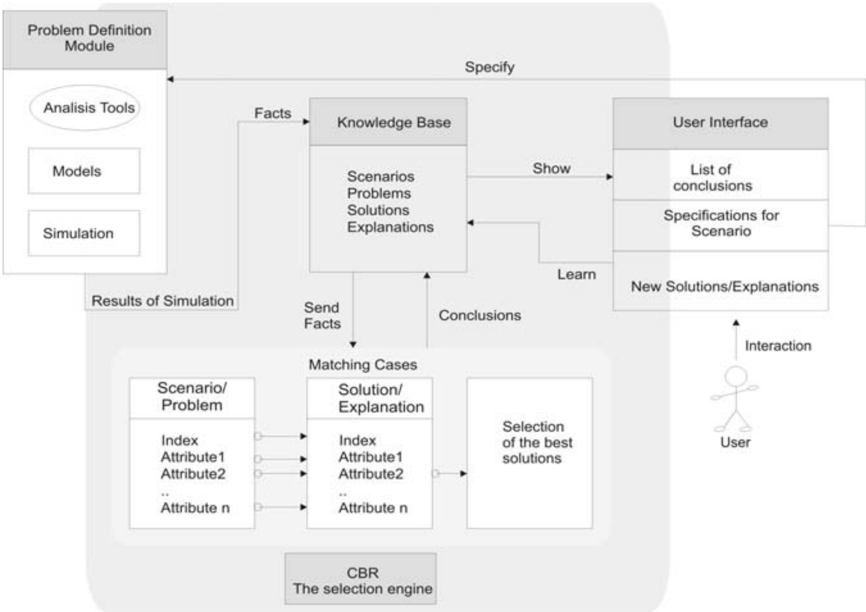


Figure 14.3. Interrelations between modules

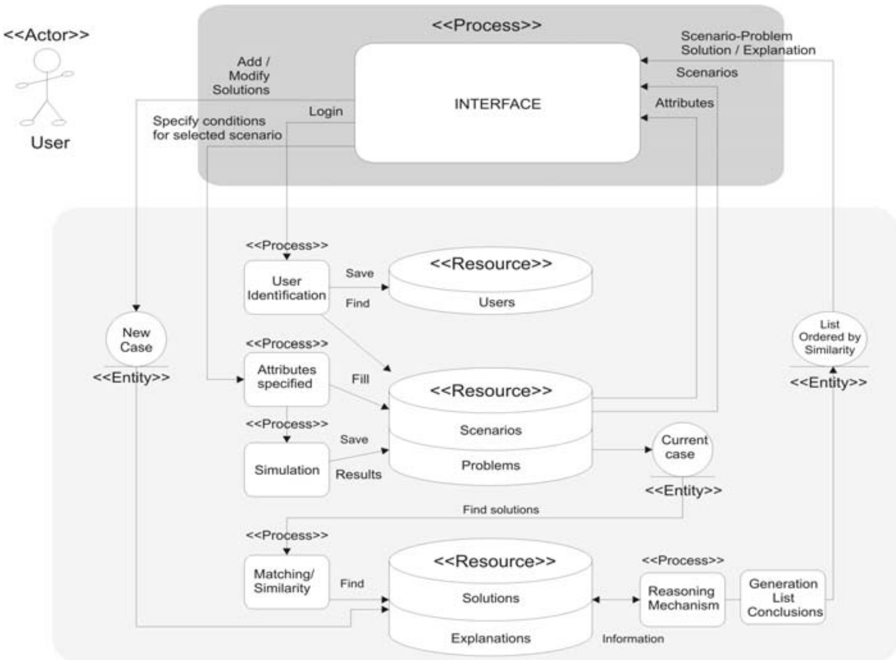


Figure 14.4. Sequence diagram:. providing solutions for a scenario/problem

14.4 The Decision-makers' Biases and how the IDSS Improves the Decision-making Process

14.4.1 How Human Beings Make Decisions

Human beings have the ability to make good decisions quickly when confronted with complex situations (Legna 2000). But they can also make bad decisions when not suitably prepared to deal with a situation; clearly, proper preparation improves strategic decision making. Our methodology aims to reduce the probability of committing strategic errors.

This section examines how our system can help political leaders and managers make better decisions. We will start by briefly reviewing some of the biases that affect decision-making processes.

As Kahneman *et al.* (1982) have demonstrated, when decision makers deal with complex situations they depend upon a few rules. These are rules that they have been refining throughout their careers as a way to simplify contexts, because they cannot do an exhaustive analysis of all the information (even when available), nor can they do a reliable cost-benefit analysis. They use their *savoir faire*, the knowledge they have acquired over time in their field of expertise.

But these rules do not always lead to good decisions. In uncertain situations requiring strategic decisions, human beings tend to be affected by cognitive biases that lead to bad decisions. Based on their laboratory experiments, Tversky and Kahneman (1974) conclude that the biases are mainly due to three heuristics: *representativeness*, *availability* and *adjustment and anchoring*. The first one is the tendency to "imagine that what we see or will see is *typical* of what can occur" (Hogarth 1980, p. 217). Availability refers to the following phenomenon: "when imagining what could happen, we remember similar past situations" (Hogarth, 1980 p. 217). The third bias is the tendency of the decision makers to formulate judgments based solely on an initial evaluation or anchor, failing to make necessary adjustments subsequently. According to these authors, each one of these heuristic can lead to bad decisions.

Since strategies are selected in an uncertain, ambiguous and unstructured context, it is unreasonable to hope that the strategists will not fall back on heuristics, leading to several cognitive biases. In fact, diverse studies confirm the tendency to fall back on them; and enlarge, in the case of strategic decisions, the conclusions of the authors mentioned in the precedents paragraphs. (Das and Teng 1999, p. 760). The following paragraphs will describe the heuristics that have empirically been found to be most frequent among decision makers (Das and 1999, p. 760)[5].

The first cognitive bias is depending upon *prior hypotheses* without later adjustment. Research shows that decision makers have a propensity to make decisions based on their beliefs and previously formed hypotheses. For example, if

---

[5] Although we define the basic heuristics following these authors, we organize them differently.

they have preconceptions regarding the relationship between some variables, they frequently tend to ignore information that contradicts them.

A second bias consists of *focusing on limited targets* instead of on all the possibilities. They tend to concentrate on objectives that for whatever reason attract their interest and to ignore others. Control systems are important, because they highlight a broader set of problems and goals. For example, budgetary control induces decision makers to concentrate their attention on the critical results from the perspective of this kind of control.

These two biases lead to an inadequate perception of the environment and of the problems that need to be solved.

The third frequently occurring cognitive bias is that decision makers *only expose themselves to a limited number of alternatives that can achieve a goal.* The decision makers pay attention to alternatives sequentially and use their intuition and emotions to complement rational analyses; but the evidence shows that they do not consider all of the possible alternatives - as was evident in President Kennedy's decisions during the Cuban Missile Crisis (Damasio 1994, Allison 1988, Baron 2000, Legna 2005).

Another important tendency that can produce pernicious effects on decision-making processes is the *insensibility to outcome probabilities.* This may be either a consequence of the tendency of human beings to commit logical errors (see the following box in green) or their ignorance of probability theory (Baron 2000, p.146-48). Empirical research shows that decision makers frequently do not trust the results of probability calculations. Moreover, they do not understand them so they do not use them. They are more influenced by possible results than by estimating the probabilities. They often prefer to use personal appraisals to describe a situation, rather than trust data resulting from the calculation of probabilities. Another reason why decision makers do not use the estimations of probabilities is that they see problems as isolated, therefore, estimating probabilities and the considering statistics on comparable past events is irrelevant.

A fifth common heuristic that produces an important bias is the *illusion of manageability* of the results of the decisions. This illusion manifests itself in two ways. First, “decision makers may inappropriately perceive a success probability higher than the objective probability would warrant” giving them "the illusion of control" (Das and Teng 1999, p.762). They do not accept the fact that risks and uncertainties are inherent in all actions. They think that because they can control the actions and their results they can reduce the risk. Secondly, the decision makers think that if problems arise during the execution they can fix them and still obtain good results. They trust in a “postdecisional control” that allows them to influence whatever goes on after their decision. “The illusion of manageability of bad outcomes eases managers' anxiety over such outcomes” (Das and Teng 1999, 763)

14.4.2 Conclusions: How the IDSS Helps to Improve Decisions and Reduce Cognitive Biases

Frequently, in governmental organizations there is not a team or DSS working to avoid cognitive biases. This is why we have designed our IDSS, which reduces the probability of falling into these biases, because it encourages:

- permanently reviewing the previous hypotheses;
- enlarging both the set of objectives and possible actions;
- the supervision of the application of the rules of the logic, especially regarding the estimation of the probabilities of the events;
- avoiding the illusion of manageability.

The results of this chapter indicate that in order to improve decisions it is also necessary to organize the decision-making processes in a way that integrates the minds of the specialists and the leaders. The former tend to emphasize the use of rational procedures, whereas the latter rely more on their intuitions and emotions. Of course this does not mean that each group uses **only** one type of thought process, but rather that in each group a certain way of understanding problems and preparing decisions **predominates**. Nevertheless, both are necessary.

In fact, our approach built the IDSS based on four pillars:

- the creation of qualitative models that allow the treatment of nonquantifiable information and variables that are crucial in order to take strategic decisions;
- the application of systems dynamics to these models, in order to understand the loops between the variables, the structure of the social system and to build possible future scenarios;
- the detection of the main strategic problems, by contrasting the possible scenarios with the operational objectives (and the scenarios that they imply);
- and the application of CBR methodology to search for possible solutions, to design and select one course of action and, finally, to learn from the social and political system's own experience as well as from policies enacted in other ones.

In other words, the IDSS may be a tool to improve the governance of the societies and quality of life of the citizens.

References

Aamodt A, Plaza E (1994) Case-based reasoning: foundational issues, methodological variations, and system approaches. AI Communications 7(19): 39-59.

Allison GT (1988) La esencia de la decisión. Análisis explicativo de la crisis de los misiles de Cuba. Buenos Aires: Grupo Editor Latinoamericano.

Baron J (1998) Thinking and Deciding. Cambridge: Cambridge University Press,.

Bernoux PH (1985) La sociologuie des organisations. Paris: Éditions du Seuil.

Damasio A (1995) Descartes' error. Emotion, Reason, and the Human Brain. New York.: Avons Books.

Das T, Teng KB., 1999. Cognitive biases and strategic decision processes: an integrative perspective. Journal of Management Studies, November: 757-779.

Hogarth RM (1980). Judgment and choice: The psychology of decision. Chichester: John Wiley & Sons.
Kahneman D, Slovic P, Tversky A. (Eds) (1982). Judgment under Uncertainty: Heuristics and biases. New York: Cambridge University Press.
Klajic M, Legna C, Skraba A (2002) Systems dynamics model development of the Canary Islands for supporting strategic public decisions. Twentieth International System Dynamics Conference, July 28-August 31. Palermo, Italy.
Klajic M, Legna C, Skraba A (2003) Simulation model of Canary Islands, Preliminary Results. 21st System Dynamics Conference, July 20-22, New York.
Legna C (2005) Gestión Pública Estratégica y Prospectiva, con aplicaciones al ámbito regional y local. Extremadura: Abecedario.
Legna C, Rivero-Ceballos J (2001) Las particularidades de las regiones ultraperiféricas y la necesidad de instrumentos específicos. Especial referencia a Canarias. Report elaborated for the Canarian Government.
Legna C (2000) El proceso de toma de decisiones públicas: ¿ racionalidad o artesanía no cartesiana ? IV Jornadas de Política Económica: La Unión Europea y el debate sobre el estado del bienestar", December 4-5 Santa Cruz de Tenerife, Spain
Riesbeck CK, Schank R (1989) Inside case-based reasoning. Northvale, NJ: Erlbaum.
Simon HA (1977) The New Science of Management Decision. Englewood Cliffs, NJ: Prentice Hall.
Turban E, Aronson J (2002) Decision Support Systems and Intelligent Systems. Seventh Edition. New Jersey: Prentice Hall.
Tversky A, Kahneman D (1974) Judgement under uncertainty: heuristics and biases. Science (185): 124-131
Yager R (1986) Decisions with usual values. In: A Jones et al (Eds) Fuzzy Set Theory and Applications. New York: D. Reidel Publishing Company, pp 115-131
Zolghadri M, Lecompte T, Bourrrieres JP (2002) Design of decision support systems for extended enterprise. Journal of Studies in Informatics and Control with Emphasis on Useful Applications of Advanced Technology, 11(1): n.a.

15

e-Negotiation Systems and Software Agents: Methods, Models, and Applications

Peter Braun[1], Jakub Brzostowski[1], Gregory Kersten[2], Jin Baek Kim[2], Ryszard Kowalczyk[1], Stefan Strecker[2], and Rustam Vahidov[2]

1 Faculty of Information and Communication Technologies, Swinburne University of Technology, Hawthorn, Australia
Emails: {pbraun, jbrzostowski, rkowalczyk}@it.swin.edu.au
2 John Molson School of Business, Concordia University, Montreal, Canada
Emails: {kersten, jbkim, strecker, vahidov}@jmsb.concordia.ca

Negotiation is a decentralized decision-making process that seeks to find an agreement that will satisfy the requirements of two or more parties in the presence of limited common knowledge and conflicting preferences. Negotiation participants are agents who negotiate on their own behalf or represent the interests of their principals. When electronic negotiations enter the stage, these agents could be intelligent software entities that take part in the process of searching for an acceptable agreement. The degree of involvement of these "intelligent agents" in negotiations can range from supporting human negotiators (*e. g.* information search, offer evaluation) to fully automating the conduct of negotiations. Choosing the degree of involvement depends upon the characteristics of the problem in the negotiation. In this chapter, we review electronic negotiation systems and intelligent agents for negotiations. Different types of negotiation agents, their roles and requirements, and various methods for effective support or conduct of negotiations are discussed. Selected applications of intelligent negotiation agents are presented.

15.1 Introduction

Negotiation is a decentralized decision-making process used to search for and arrive at an agreement that satisfies the requirements of two or more parties in the presence of limited common knowledge and conflicting preferences. Negotiation processes appear in a multitude of forms. They occur in very different situations and are influenced by ethical, cultural and social circumstances. These processes and their participants have been a research topic of many disciplines including anthropology, psychology and sociology, political sciences (Ury 1993; Fisher *et al.* 1994), law (Wetlaufer 1996), economics (Young 1975, Roth 1995), applied mathematics (Harsanyi 1997), and computer science (Sycara 1997, Kraus 2001).

The use of software to support negotiation processes was put forward in the late 1970s. Empirical research on computer-mediated communication systems such as Hiltz and Turoff (1978) preceded research on systems supporting negotiations. Keen and Scott-Morton (1978), Sprague and Carlson (1982), and others proposed to extend decision support system (DSS) capabilities to aid the negotiators. This led, in the early eighties, to the design of negotiation support systems (NSSs) and group decision support systems (GSSs) (Korhonen *et al.* 1986, Jarke *et al.* 1987, Jelassi and Foroughi 1989). Negotiation support systems are designed to help and advise negotiators during the various phases of the negotiation process; they are used to structure and analyze the negotiation case, elicit preferences and use them to construct a utility function, determine feasible and efficient alternatives, set negotiation tactics, visualize different aspects of the problem and the process, and facilitate communication.

NSSs are based on the modeling approaches formulated in decision sciences, negotiation analysis, and game theory (Raiffa *et al.* 2003). The contribution of decision science to negotiation includes decision rules, decision trees, single- and multi-attribute utility theory, and statistical methods such as forecasting and regression analysis.

Negotiation analysis integrates decision analysis and game theory in order to provide methodological support to negotiation participants. Approaches based on negotiation analysis aim at bridging the gap between descriptive behavioral models and normative formal models of bargaining. These approaches have adopted a number of behavioral concepts, including reservation and aspiration levels, the best alternative to the negotiated agreement (BATNA), and integrative and distributive negotiations, and incorporated these concepts into quantitative models (Kersten 2001). This allowed advisors to conduct formal analysis of negotiations in order to support negotiators. Other approaches stemming from computer science, especially Artificial Intelligence, have also been used in the design of software that aids one or more negotiators (Matwin *et al.* 1987, Rangaswamy *et al.* 1989, Kersten 1993).

A good classification of NSSs and DSSs can be found in Starke and Rangaswamy (1999) who distinguish them by preparation and evaluation systems and process support systems, and in Kersten (2004) who further classifies them considering the phase of the negotiation process: (1) planning systems; (2) assessment systems; (3) intervention systems; and (4) process systems.

The Internet and new computing and communication technologies introduced new opportunities for the design and deployment of software capable of supporting negotiators, mediators and arbitrators. Negotiations conducted over the Web are commonly called e-negotiations and the systems used in e-negotiations are named e-negotiation systems (ENSs). ENSs are information systems that employ Internet technologies that are deployed on the Web. Defining ENSs as software deployed on the Web, capable of aiding one or more negotiators, mediators or facilitators allows us to include e-mail, chat and streaming video used in negotiations (Moore *et al.* 1999, Lempereur 2004), as well as software used for automated negotiations and auctions (Zlotkin 1996, Jennings *et al.* 2001).

e-Negotiation systems are unlike previous systems deployed on standalone computers or local- and even wide-area networks in terms of the implemented mechanisms and employed technologies. Specifically, the potential of intelligent software agents has been noted for their suitability in a distributed computing environment such as the Internet.

Software agents are programs that carry out certain operations on behalf of a user or another program with some degree of independence or autonomy and, in doing so, realize a set of goals or tasks for which they are designed (Jennings and Wooldridge 1998; Maes 1998; Jennings 2001). These programs differ from regular software because they are personalized, continuously running, and to a certain extent autonomous. The reasoning mechanisms of software agents can range from a set of simple "if-then" rules to sophisticated machine learning algorithms such as neural networks or Bayesian networks (Caglayan and Harrison 1997, Wooldridge 1999).

Software agents carrying out negotiation activities on behalf of users are known as negotiation software agents (NSAs). These agents have been developed to study the automation of different negotiation tasks that arise from buying and selling products over the Internet.

The purpose of this chapter is to investigate the full potential of NSAs and explain research issues. To do so, we first review e-negotiation systems and investigate models for positioning NSAs in ENSs. Then, we examine models and techniques for NSAs. Finally we present applications of NSAs, and conclude with an outlook to further development.

15.2 Foundations

The use of software in negotiations requires that a *process model* and a *protocol* is constructed (Kersten and Lo 2003, Kim and Segev 2003). The process model describes negotiation phases and assigns different activities to them. Its significance is in that it allows the negotiators to follow a methodologically sound approach (Lewicki *et al.* 1999). The protocol is a formal model, often represented by a set of rules, which governs software processing and communication tasks, and imposes restrictions on activities through the specification of permissible inputs (Jennings *et al.* 2001).

Behavioral research on negotiations has so far not included the processes in which support systems and software agents are involved as active participants and, therefore, no process models have been developed specific to e-negotiation.

Hence, we need to adapt a behavioral phase model to reflect the requirements imposed by an ENS. We have adapted a model proposed by Kersten (1997), which is based on Gulliver's eight-phase model (1979). This model has been modified to allow for a wider range of negotiated decisions than the eight-phase model, including those which use ENSs. The model, presented in Figure 15.1, comprises the following five phases:

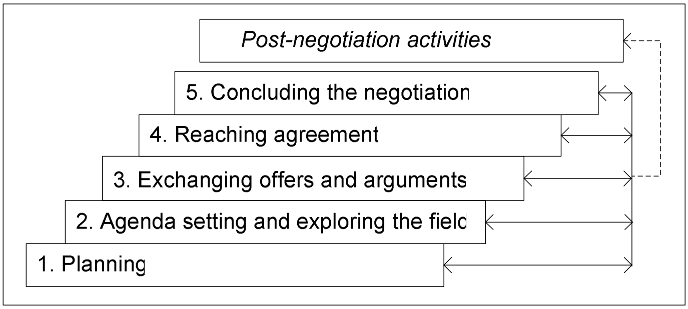


Figure 15.1. Negotiation process model

- The planning phase comprises activities that the negotiators undertake both individually and jointly. They formulate their representation of the negotiation problem including the specification of issues and options. In this phase the negotiators specify their objectives and preferences, and such negotiation-specific constructs as BATNA and reservation levels (Fisher *et al.* 1991). If the negotiators know or can learn about their opponents, they decide on strategies to be used. This phase's joint activity also includes the selection of the negotiation location and time, and the communication modes the negotiators will use.

- Agenda setting and exploring the field includes the negotiators' discussions about the negotiated issues and their meaning. The discussion's result may be that new issues and options are added or some may be deleted. The negotiators may also discuss the protocol they will follow, the timing of the exchanges, the deadline and—in some negotiations—their objectives, priorities and constraints. The result of these discussions is that the negotiators may have to revise the problem, objectives and preferences, and also their strategies and initial tactics.

- Exchanging offers and arguments allows the parties to learn about the others' limitations, and to identify the key issues and critical areas of disagreement. During this phase, the parties realize the potential of a compromise and can assess its main features. The analysis of a negotiation may focus on the modification of strategies, the determination of concessions and revision of aspiration levels, and on the restriction of efficient solutions to those that may be acceptable to the parties.

- Reaching an agreement means that the parties realize that the negotiation has been successful. Having identified the critical issues, they may develop joint proposals or soften their individual limitations. The parties may also identify a limited number of possible compromises.

- Concluding the negotiation takes place when the negotiators reach an agreement. They evaluate this compromise and consider its possible improvements. They also may discuss additional issues that, however, have no impact on the negotiations (*e. g.* the agreement implementation).

The negotiation phase model provides a structure to the negotiation process. This is not to say that the negotiator who conducts activities belonging to one phase cannot return to another. Negotiations rarely proceed in a linear fashion. While the negotiation methodology suggests that the parties should not bypass or enter a phase before completing the previous phase, the parties may in reality at some point during the course of the negotiation need to return to one of the previous phases. This situation could arise if information obtained during the process requires a revision of assumptions and/or specifications. For example, parties may suggest additional communication channels and propose new issues to be negotiated. Such changes may require the revision of reservation levels and of preferences. This possibility of revisiting previous phases and then returning to the current phase is indicated in Figure 15.1. The possibility of ignoring some phases is also shown in Figure 15.1. This practice, while it occurs in real-life negotiation, should not be allowed in ENS-supported negotiations. Instead, the process model and protocol underlying an ENS should be based on a sound methodology and carried out, among others, by the sequencing imposed by the phase model.

15.3 e-Negotiation Systems

The use of the early NSSs was limited due to: (1) limitations of information and communication technologies (ICTs); (2) limited computer literacy of managers, who, therefore, delegated the system usage to analysts; (3) complexity of the constructed models, often based on strong rationality principles that required a significant amount of users' input; and (4) insufficient consideration of psychological and sociological conditioning of negotiations.

New ICTs, including Internet, software architectures, and software development technologies made rapid development of systems for millions of users possible. They also exposed people to information systems and their practical use for shopping, communication, information retrieval and entertainment. These developments led to two new streams of theoretical and applied research:

- Behavioral research on the use of communication technologies (mainly email) in negotiations (Purdy and Neye 2000, Thompson and Nadler 2002); and

- Design and development of easy-to-use systems for negotiation support (Kersten 1999).

e-Negotiation systems (ENSs) have one or more of the following capabilities:

- To support decision and concession making;
- To suggest offers and agreements;

- To assess and criticize offers and counteroffers;
- To structure and organize the process;
- To provide information and expertise;
- To facilitate and organize communication;
- To aid agreement preparation; and
- To provide access to negotiation knowledge; experts, mediators or facilitators.

Early ENSs were designed by university researchers around a single model or a procedure. The Inspire system (Inspire 1996) is an implementation of a method for utility construction based on hybrid conjoint analysis (Kersten 1999). The Joint Gains system (JointGains 2000) implements the joint improvement directions method (Ehtamo *et al.* 2001). Since the mid-1990s these two systems have been used in teaching and research. The virtual property agency system (defunct) implemented linear weighted value function (Bui *et al.* 2001). Some of these systems do not support negotiations but facilitate them by providing an electronic bargaining table (Rangaswamy and Shell 1997). Other systems focus on the contracts' preparation and the content of documents (Schoop and Quix 2001).

Most of the early commercial ENSs were also single-purpose. CyberSettle (Cybersettle 2000) is an online system that supports its users in negotiating single-issue insurance claims. It has a simple conflict resolution mechanism based on expanding offers made by each party by 20%. The Electronic Courthouse (NovaForum 2000) provides alternative dispute resolution (ADR) services by linking claimants with a roster of lawyers and ADR professionals.

The common feature of these systems is their orientation to one specific type of negotiation interaction. They provide a service that is based on the assumptions underlying the theoretical model or—in the case of commercial ENS—the implemented business rules. These types of systems are *ENS applications* supporting one type of negotiation (Rebstock 2001, Neumann *et al.* 2003). Table 15.1 summarizes ENS functions and activities.

NSSs have very limited autonomy; their purpose is to provide support to one or more users to assess decision alternatives, select offers and evaluate counteroffers, and to communicate with their counterparts. In contrast, NSAs have significant autonomy in their decision-making and communication activities. The NSA acts for and on behalf of the principal, the agent actively helps the principal and seeks information, evaluates the principal and others decisions, and communicates with the counterpart. While both an ENS and NSA may try to help the negotiators understand the problem, express their preferences, represent the process and formulate the exchanges, an ENS is passive; it does not attempt to seek information from various sources, interfere in the process, propose and/or make offers, or assess and present arguments for offer acceptance or rejection.

Table 15.1. ENS functions and activities

Function	**Activities**
Communication, presentation and interaction	
Transport and storage	Transport of information among heterogeneous systems; storage in distributed systems; security.
Search and retrieval	Search of information; selection; comparison and aggregation of distributed information.
Formatting, presentation and interaction	Data formatting for other systems use; data visualization, alternative data presentation, user-system interaction.
Modeling and content formulation	
Decision problem formulation	Formulation and analysis of the decision problems; feasible alternatives; decision space, measurement.
Decision-maker specification	Specification of constructs describing decision makers; preferences; measures for alternative comparison; negotiators' models and styles.
Strategies and tactics	Evaluation and selection of the initial strategies and tactics.
Negotiation	
Offer and message construction and evaluation	Formulation of offers and concessions; analysis of messages and arguments; argumentation models.
Counterpart analysis	Construction and verification of models of negotiation counterparts; evaluation and prediction of their behavior
What-if, sensitivity and stability analyses	Analysis of offer and counteroffer implications; analysis of the implication of different offers on the counterparts' reactions; assessment of the potential compromise solutions.
Process, history and their analysis	Construction of the negotiation history; process analysis; progress/regress assessment; history-based predictions.
Knowledge seeking and use	Access and use of external information and knowledge about negotiation situations and issues arising during the process; comparative analysis.
Negotiation protocols	Specification of, and adherence to, the negotiation agenda and rules
Strategies and tactics	Assessment counterparts' of strategies and tactics; modification of strategies and tactics

The possible functions of NSAs depend—in addition to the available technologies and knowledge—on their required degree of the agent's autonomy, the type of negotiation, and the specificity of the principal's instructions. The functions depend also on the expected scope and form of the agent's interactions with other systems and agents. The agent may be highly specialized and may cooperate with other agents, interact directly with the principal, or it may communicate via a DSS or a NSS that supports the negotiators in the construction of problem representations and in their assessment and modification. The agent may follow the principal directives or it may suggest new issues/options and innovative (for the principal) approaches to cope with conflict, based on the information obtained from experts and extracted from other negotiation histories.

Negotiation software agents may take over well-defined and structured activities in a negotiation but it is not necessary for the agents to handle all of the tasks. For example, the agent may present offers, seek information about similar negotiation situations, collect information about the counter-parts, and alert the principal if pre-defined conditions are violated. The ill-defined and ambiguous issues, decisions regarding relationship between the parties, modification of the rules and parameters are better left to the principals.

Complex and rich processes comprise both routine and simple tasks as well as tasks that are original and require imagination. Business negotiations are often an example of processes requiring the use of both ENS and NSA technologies. There is a need to develop tools and infrastructure that can support and also independently conduct activities. In business-to-business negotiations flexible and extensible tools are needed to support both integrative and distributive activities. These tools have to be highly interactive and competent at managing the complexity of multilateral business-partner relationships, especially since each business negotiation tends to be unique, in small, but important, ways.

Among other things, a particular architecture depends on the complexity of interactions with the principal, level of support required, and the requirements for information processing by other systems (*e. g.*, financial, marketing and production). In the next section, we explain the types, functions, and architectures of NSAs for ENSs.

15.4 Negotiation Software Agents

This section aims at synthesizing the relationships between software agent capabilities and relevant tasks in different negotiation phases within a coherent framework. Despite the lack of a well-formulated and widely accepted definition of the concept of software agent (Wooldridge and Jennings 1995, Franklin and Graesser 1997), we adopt a natural metaphor view of an agent (Jennings and Wooldridge 1998). This allows us to use the notion of an agent as an abstraction tool for structuring the design of complex software systems (Jennings 2000, Jennings 2001, Luck *et al.* 2003).

The first important issue to be addressed is what types of agents will be useful in supporting negotiation tasks. Franklin and Graesser (1997) have proposed a classification scheme for agents based on the properties they possess. Nwana and Ndumu (1998) have identified autonomy, cooperation, and learning as subsets of dimensions for deriving classes of agents. In their schema, agents possessing cooperation and autonomy features would be referred to as “collaborative agents”, while those with learning and autonomy properties would be described as “interface agents”. Agents possessing all three features were identified as “smart” agents.

Kinny *et al.* have advocated the methodology for belief-desire-intention (BDI) agent-based systems that includes identifying the roles and duties of these agents as the initial step (Kinny *et al.* 1996). We will follow a similar approach here by first proposing the type of agents to be potentially employed in support of e-negotiations, and then detailing the responsibilities they will have in regard to different phases of the negotiation process. A similar approach was used in studies on the design of agent-based decision support systems. There, the authors identified groups of agents supporting the three phases of Simon's problem solving model (Simon 1977), including intelligence, design, and choice (Vahidov and Fazlollahi 2004).

In order to conceptualize the role of agents in e-negotiation support, it would be useful to think of the negotiation situations along two dimensions. One relates to the willingness of the negotiators to disclose their private preferences to a third party, which promises to make the negotiation process more efficient. The other one relates to the degree of certainty regarding negotiator preferences and strategies (*i. e.* the degree to which the negotiator's task can be regarded as being "structured"). The types of agents that specifically suit e-negotiation tasks are described below.

- *User profile agent.* The purpose of this type of agent is to elicit user preferences, and to assist the negotiator in deciding on objectives and strategies. Ideally agents of this type would be able to adapt to the changes in user behavior in the process of negotiations.

- *Information agent.* Agents of this type would engage in actively seeking, retrieving, filtering, and delivering information relevant to the issues on the table.

- *Opponent profiling agent.* The primary purpose of this agent type would be to identify the objectives, preferences and strategies of the opponent. Knowing the opponent better renders offer generation and evaluation a much better informed decision-making process. The information and opponent profiling agent could be regarded as "intelligence" agents.

- *Proposer agent.* The aim of this type of agent is to generate a set of promising offers to be considered for submission to the opponent. In negotiation problems that involve multiple issues, the generation of an offer may involve search in a very large space of possible offers.

- *Critic agent.* The purpose of the critic is to evaluate the offers received from and addressed to the opponent and provide "verbal" feedback on the drawbacks and, possibly benefits of these offers. The proposer and critic agents could be regarded as a type of "adviser" agents.

- *Negotiator agent.* This agent may be capable of conducting negotiations by itself in a semiautonomous or fully autonomous fashion. Applicability of full automation depends on the degree of certainty in objectives, preferences, and tactics of the negotiator (*i. e.* the level of structuredness of the negotiation task from the negotiator's perspective).
- *Mediator agent.* The main purpose of this agent is to coordinate the activities of the negotiating parties, and to attempt to generate mutually beneficial offers. The role of this agent increases when the parties are willing to provide their information to a third party agent.

In order to identify the tasks that could be potentially delegated to agents we have to revisit the phase model of negotiations and find out which activities are potentially amenable to automation. The major activities of the planning phase include formulation of negotiation problem, including the specification of issues, options, BATNA, reservation levels, and negotiation strategies. Any important knowledge about the opponent would help to better prepare oneself for negotiations. Agents can assist the negotiators by finding information about the markets related to negotiations (recent deals, prices, *etc.*), capturing user preferences (*e. g.* through use of such techniques as conjoint analysis), helping the negotiator define the adequate negotiation strategy, and profiling an opponent (inferences about opponent based on background information).

The second phase includes agenda setting, exploring the domain, and discussion of negotiated issues, negotiation protocol, timing of exchanges and the deadline. Agent contributions are limited in this phase as this involves mostly human-directed activities. One possible application is for the mediator agents to obtain negotiator preferences, try to match them and send messages to negotiators about the acceptable set of issues, time, deadline, *etc.*

Exchanging offers and arguments and, possibly reaching an agreement are the main "action" phases in negotiations. In these phases, information agents deliver up-to-date information to advise the negotiator about market prices, deals, *etc.* A critic agent can evaluate offers received from the component and provides its opinion to the negotiator on the acceptability of offers. It would also watch over the shoulder of the user when the user prepares an offer. This agent can interfere to criticize the offer in regard to its alignment with user's interests, strategy, and current market situation.

The proposer agent would help formulate offers by generating a set of different promising offers in accordance with current preferences and strategy selected by the user. A user-profile agent would adjust the user profile by watching the actual offers chosen or formulated by the user. Opponent profiling agents likewise would update the opponent profile by watching opponent's moves. The mediator may also watch the exchanges and learn about the parties' preferences. The mediator agent could get involved if parties agree to submit their preferences to the mediator that would do the matching and generate candidate agreement packages to the negotiators. Finally, if included, a negotiation agent could take over the negotiation process itself and interact with the opponent in an autonomous mode.

Table 15.2. Agent-supported tasks

Agent type	Negotiation phases			
	Planning	**Agenda setting**	**Conducting**	**Concluding**
User Profile	Eliciting user preferences, helping with the choice of a strategy		Tracking user behavior, maintaining user preferences	Updating and storing user preferences
Information	Delivering relevant information for planning		Delivering latest information relevant to an ongoing exchange	
Opponent profile	Deriving an initial profile of an opponent		Tracking and updating opponent profile	Updating and storing opponent profile
Proposer			Generating promising candidate offers	
Critic			Evaluating and critiquing offers and counteroffers	
Negotiator			Conducting negotiations (well-structured tasks)	
Mediator		Coordinating negotiation issues, protocols, settings	Generating set of potentially acceptable agreement alternatives (private information disclosed)	Offering possible improvements

In the concluding phase of the negotiations the mediator agent may analyze the estimated utility of an agreement and perhaps propose a few more alternatives to the negotiators if it finds potential room for improvement. Table 15.2 summarizes support provided by agents in different phases of negotiations.

Figure 15.2 shows the generic architecture for an agent-enhanced e-negotiation system. Analytical models and local data are included in the “toolbox” part of the system. These tools are used by the variety of agents in order to carry out their tasks. Information retrieved by the information agent can be used by different agents to assist in setting objectives, generating alternatives, and critiquing offers. Most of these activities could be also assisted by opponent profile information.

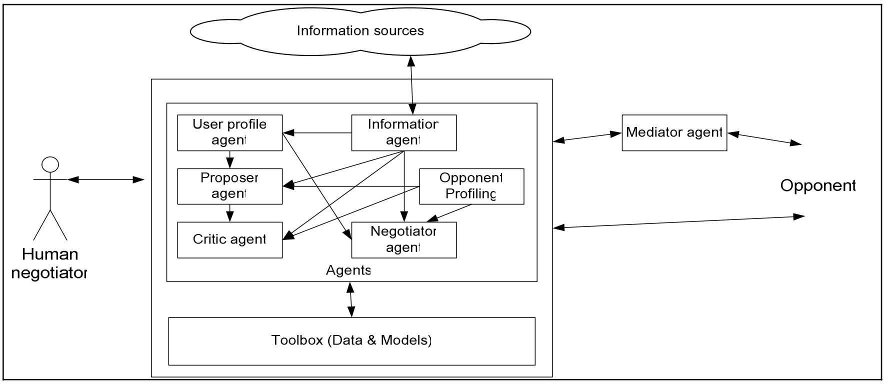


Figure 15.2. Generic architecture for agent-enhanced e-negotiation system

15.5 Models for Negotiation Software Agents

In this section, we describe several models proposed for negotiation software agents over recent years. They vary from decision-making models of negotiation to learning methods for supporting the negotiation, based on a variety of techniques including: probabilistic decision theory, possibilistic decision theory, Bayesian learning, possibilistic case-based reasoning, constraint-based reasoning, heuristic search, Q-learning and evolutionary computation.

Various techniques can be applied to automate the negotiation process and its selected activities. The first approaches were based on game theory. However, game theory makes a number of assumptions including knowledge of circumstances (Jennings 2001). This means that we should know rules of the encounter, specify our preferences, and know our partners’ preferences or at least be able to formulate beliefs about their preferences. Another assumption of game theory is the full rationality of negotiators, which means that agents have sufficient reasoning and computational capacity to maximize their expected payoffs given their beliefs. Because of these drawbacks, we do not consider game-theoretic approaches in the remainder of this section.

In general, negotiation software agents require an appropriate protocol, specification of negotiation objects, and an apparatus for decision making. An interaction protocol (Binmore and Vulkan 1999) defines rules of negotiation, for example, who is allowed to say what and at which time. It defines roles and actions that negotiators can take at each moment of the negotiation process. Negotiation is usually a multistage process. Some protocols allow agents to submit proposals simultaneously, for example the monotonic concession protocol (Rosenschein and Zlotkin 1994). Other protocols allow for iterative exchange of proposals, for example the iterative negotiation protocol (Stahl 1972). In both types of protocols, an agreement is reached if one agent matches what the other one asked for, that is the agents agree on the same terms.

The object of negotiation may be a service or commodity and it is usually characterized by a number of attributes that are also called negotiation issues. The operators that can be performed on the attributes are defined on these objects, for example to change their values and add new issues to a negotiation.

The decision-making apparatus defines a model for decision making and strategies used by an agent during negotiation. Depending on what knowledge is available and the circumstances of the particular encounter, different apparatuses for decision making can be designed.

Real-life negotiation problems are typically ill-defined, information is not equally distributed among the participants, the participants have only partial knowledge about their counterparts and communication is often ambiguous or imprecise. Methods of artificial intelligence (AI) are particularly useful in problems, in which knowledge about partners' types and full rationality cannot be assumed. In such approaches negotiation agents can use AI-based decision-making mechanisms that satisfy the bounded information, bounded rationality, and bounded computational characteristics of agents (Binmore 1992, Rubinstein 1998). The lack of knowledge about partners' types can be compensated by the agents' ability to learn about and verify the acquired information. Agents need to be able to update their knowledge about their partners as well as the environment. This capability is the prerequisite for negotiating agents to be able to adapt their behavior to cope with changing partners and changing user preferences.

Given the negotiation problem two kinds of knowledge can be distinguished: knowledge of the opponent and knowledge of oneself. Considering approaches coming from game theory, knowledge of the problem can be reduced to the payoff tables. Knowledge about the opponents is required in order to be able to determine the moves (offers) they may accept and those they reject.

The acquisition of information such as counterpart's preferences, reservation price, or deadline, *etc.* allows us to increase knowledge about the partners. The second kind of knowledge is knowledge of oneself, which is used to select one strategy from the space of all possible strategies during the process of negotiation.

Strategy selection depends on the negotiator's objectives, preferences, and risk attitude. For a strategy to be effective it has to lead to a solution, which the negotiators' counterparts accept. This means that in constructing the set of possible strategies the counterparts' profiles have to be considered. The question that arises is: "How to learn about partners to devise adequate strategies?" Usually there are two kinds of information available, that is, the history of previous interactions and the behavior of an opponent during the current encounter. There is no universal method known for handling the available data to learn about partners' preferences in order to derive optimal moves during the negotiation process. Researchers apply and test various models of data acquisition and inference (Gerding *et al.* 2004). Below we present several models of decision-making and AI-based learning approaches for supporting negotiation. The models include probabilistic decision theory, possibilistic decision theory, constraint based reasoning, and heuristic search. The learning approaches for supporting the negotiation include: Bayesian learning, possibilistic case-based reasoning, Q-learning, and evolutionary computation. Table 15.3 summarizes the described approaches.

Table 15.3. Summary of negotiation approaches

Approach	**Characteristics**
Probabilistic DT	Selecting optimal decision
Possibilistic DT	Selecting optimal decision
Bayesian learning	Learning negotiation partner's type
Possibilistic CBR	Selecting most prospective negotiation partners
Constraint-based reasoning	Finding a solution that satisfies constraints of negotiating partners
Heuristic search	Determination of negotiation offer
Q-learning	Searching the set of potential strategies
Evolutionary computing	Searching the set of potential strategies

15.5.1 Probabilistic Decision Theory

In this model the uncertainty about the consequences of a decision $\boldsymbol{d}$ is modeled by a probability distribution $\boldsymbol{p_d} : \boldsymbol{S} \rightarrow [0,1]$ that assigns to each possible state a probability value. The decision makers preferences are encoded by the utility function $\boldsymbol{u} : \boldsymbol{S} \rightarrow [0,1]$ If the probability distribution is constructed for each of the possible decisions, then the expected utility can be calculated for each of these decisions (von Neumann and Morgenstern 1944):

$$EU(d) = \sum_{x \in S} p_d(x) u(x)$$

To maximize its outcome, an agent chooses the decision with the highest expected utility.

15.5.2 Possibilistic Decision Theory

In some situations the given information may not be sufficient to build a probability distribution. One of the alternative approaches is to employ possibility theory (Dubois and Prade, 1996). The basic notion of possibility theory is that of a possibility distribution, which is a counterpart of a probability distribution in the classical approach. The possibility distribution $\boldsymbol{\pi_d} : \boldsymbol{S} \rightarrow [0,1]$ assigns to each possible outcome a level of plausibility. The difference between probability and possibility is that the first notion is usually a measure of frequency of occurrence of an event whereas the second one is the measure of extent to which an event may occur. As in the previous approach the agent's utility function $\boldsymbol{u} : \boldsymbol{S} \rightarrow [0,1]$ also needs to be specified. The optimal decision can then be chosen according to the optimistic and pessimistic criteria

$$QU^{-}(\pi \mid u) = \min_{x \in S} \max(n(\pi_d(x)), u(x))$$
$$QU^{+}(\pi \mid u) = \max_{x \in S} \min(\pi_d(x), u(x))$$

15.5.3 Constraint-based Reasoning

Negotiation can be considered as a distributed constraint satisfaction problem (CSP) where the constraints of each party are partitioned between agents that exchange the coordination information in order to find a solution that satisfies all the constraints (Kowalczyk and Bui 2003, Sathi and Fox 1989, Yokoo 1998). The negotiation agents iteratively exchange their preferred solutions in the form of offers, and relax their preferences and constraints according to typically heuristic negotiation strategies, until all the constraints are satisfied and an agreement is reached.

The classical CSP considers constraints that can be precisely defined and fully satisfied (Yokoo *et al.* 1998), which may limit its applicability in many real-world negotiation problems, where preferences and constraints are imprecise and soft. These assumptions can be softened with a notion of fuzzy constraints that allow one to express the degrees to which the constraints are satisfied with different solutions and can be used to uniformly represent the constraints, preferences and objectives of negotiating parties (Kowalczyk 2000, Kowalczyk and Bui 2003, Kowalczyk 2002, Luo *et al.* 2003a, Luo *et al.* 2003b).

Fuzzy constraints are considered as fuzzy relations over and between the negotiation issues, and are represented by membership functions defining the degree of constraint satisfaction with the issues instantiations (possible agreements). An assignment satisfies a constraint fully if it is evaluated to 1 and violates a constraint when it is evaluated to 0. The intermediate values represent the degree of partial constraint satisfaction. For example, a fuzzy relation representing the constraints of an agent can be defined as follows:

$$C^j(x^j) = \bigwedge_{k=0,\dots,m_j} C_k^j(x^j)$$

Where $\boldsymbol{C^j(x^j)}$ is a fuzzy relation corresponding to the constraints $\mathbf{C^j} = \{ \mathbf{C^j_k} \}$, k = 1,.., $\boldsymbol{m_j}$ of the j^{th} agent, $\wedge$ is a conjunctive combination operator (*e. g.* a t-norm in the form of the min operator). The agents search for an agreement that satisfies the constraints of all the agents, that is:

$$C(x) = \bigwedge_j C^j(x),$$

through the exchange of their preferred solutions according to the level of constraint satisfaction. The search is typically guided by the negotiation strategies of each party, that is the rules for generation of offers (*e. g.* trade-off and/or concession) taking into account the information available to an agent including the individual preferences, constraints and objectives as well as the previous offers and counter-offers. The principles of fuzzy constraint-based reasoning can assist the search process by ordering and pruning the search spaces of the parties, and finding a solution that maximizes the agent's level of satisfaction subject to its acceptability by other agents (Kowalczyk 2002).

15.5.4 Heuristic Search

The strategic reasoning of a negotiating agent is usually computationally intractable. In such situations it can be supported in the search for the best strategy by some heuristic approaches.

Faratin *et al.* (1998) suggest approximating the rational choice of negotiation strategies with the use of decision functions. The idea is based on notions of heuristic strategies and tactics, which can be used by agents to calculate good proposals or counterproposals during the negotiation. The mechanism allows for the generation of offers and counteroffers in a responsive way, linearly combining simple functions that are called tactics. There are several factors substituting full rationality, which have to be taken into consideration during the negotiation. The agent has to consider deadlines and has to adapt to remaining time or any other resource constraints. The agent also has to be responsive, which means that it has to take opponent's behavior into account and this may be achieved by its imitation. Various levels of importance can be defined for different factors and these levels may also change during the negotiation because of agent's adaptation. Three types of negotiation tactics are distinguished: time-dependent tactics, resource-dependent tactics, and behavior-dependent tactics.

The time-dependent tactics are modeled by the negotiation deadline. The closer the deadline, the faster an agent concedes. A further distinction in this set of tactics can be achieved by varying function parameters. In the Boulware tactics model, the agent maintains the offered value most of the time and concedes up to the reservation value when approaching the deadline.[6] The conceder tactics are the opposite of the above tactics. Here, agents concede much in the beginning and quickly approach the reservation value.

The resource-dependent tactics are generated by the same family of functions as time-dependent tactics and constitute a broader set of tactics. Resource-dependent tactics can model any kind of resource, for example the number of negotiating agents.

Behavior-dependent tactics are the third group of tactics. These types of tactics are concerned with responsiveness to a partner's behavior during the negotiation and may be interpreted as an implementation of cooperation. They can imitate opponent's behavior in a variety of ways.

For each type of tactic there is a corresponding negotiation decision function that is used to calculate the value of the next concession. A decision function determines what should be the next offer taking into account a particular tactic. In a multi-issue scenario the decision functions can be considered separately for each issue. In some

---

6 This tactic has been introduced by Lemuel Boulware, Vice President of General Electric. After making an assessment of union strength in each organized facility, the company presented the same offer to all locals and resisted making any changes with the "take-it-or-leave" approach. Next, Boulware would approach the weakest local union and slightly improve the offer. After the weak union settled other unions were pressured to accept the same offer. They agreed on Boulware's offer because a single union could not organize a successful strike.

applications time can be the most important factor (Fatima *et al.* 2002). A negotiation decision function, $\boldsymbol{F^a}$ may be defined as follows:

$$F^a(t) = k^a + (1 - k^a)(\frac{\min(t, T^a)}{T^a})^{\frac{1}{\varphi}}$$

where: $\boldsymbol{k^a}$ is a constant for agent $\boldsymbol{a}$ determining the value of attributes to be offered as the first proposal, $\boldsymbol{T^a}$ is the deadline of agent $\boldsymbol{a}$ and $\boldsymbol{\varphi}$ determines the type of time-dependence (Boulware: $\varphi < 1$, conceder: $\varphi > 1$).

The offer proposed by an agent a to an agent $\hat{a}$ can be determined using the following formula:

$$p^t_{a \to \hat{a}} = \begin{cases} P^a_{\min} + F^a(t)(P^a_{\max} - P^a_{\min}) & \textit{for} \quad a = b \\ P^a_{\min} + (1 - F^a(t))(P^a_{\max} - P^a_{\min}) & \textit{for} \quad a = s \end{cases}$$

where $[P^a_{\min}, P^a_{\max}]$ is the range of attribute, b denotes buyer and s denotes seller.

15.5.5 Bayesian Learning

The Bayesian learning model enables updating the knowledge or beliefs of one agent about other agents (Zeng and Sycara 1997, Zeng and Sycara 1998). Before negotiation starts an agent acquires knowledge. This knowledge consists of the information about the environment and information about other players, which can be gained from various sources, such as previous experience, second-hand knowledge, or rumours. In the Bayesian approach this knowledge is encoded in a form of subjective probability distributions. We can have beliefs about environment parameters such as product supply and demand, or interest rates. As far as other players are concerned we can have beliefs about their utility function, reservation prices, deadlines or even negotiation style.

During the negotiation encounter, our agent has to use this feedback by updating its subjective beliefs about the environment and other players after every move of other participants. This stage is performed using the Bayesian rules. First, prior knowledge about the probabilities of hypotheses $\boldsymbol{H_i}$ is given ($\boldsymbol{i=1,\ldots,n}$). In other words, we have a prior probability distribution over the set of hypotheses. Then we need a new piece of evidence (new event denoted e) that can be derived from the action performed by other agents. Also, some conditional probability is needed stating how likely an event $\boldsymbol{e}$ is to occur given that the hypothesis is true. We update the prior distribution and obtain a posterior distribution according to the formula as follows:

$$P(H_i \mid e) = \frac{P(H_i)P(e \mid H_i)}{\sum_{k=1}^{n} P(e \mid H_k)P(H_k)}$$

where $\boldsymbol{P(H_i \mid e)}$ is the new probability of the hypothesis $\boldsymbol{H_i}$ assuming that a new event $\boldsymbol{e}$ occurred. Having the updated distribution we can perform the next stage which is the best action selection. The action may be an acceptance or an appropriate counter-proposal that maximizes the expected utility given the information available at this stage

The Bayesian framework enables the modification of a given subjective probability distribution during the negotiation but does not give an answer to the question of how to obtain the distribution. Classical statistical methods of constructing the distribution may be applied but it usually requires a large amount of data. Instead of the quantitative method, a qualitative paradigm may be employed to address this problem. An example of such a paradigm is the possibility-based case-based reasoning.

15.5.6 Possibilistic Case-based Reasoning

This method allows for obtaining the opponent's likelihood towards agreement in the form of possibility distribution based on past experience. The reasoning from previous cases may be performed through a possibilistic rule stating that: "The more similar the situations are the more possible are similar outcomes." (Dubois *et al.* 1998). This can be expressed by the following formula:

$$\mu(y) = Max_{(s^i, o^i) \in H} S(s^t, s^i) \otimes P(o^i, y)$$

where $\boldsymbol{S}$ and $\boldsymbol{P}$ are the similarity relations for situations and outcomes respectively, $\otimes$ is the $\boldsymbol{T}$-norm[7], $\boldsymbol{H}$ is the history of previous cases, $\boldsymbol{s^t}$ is the current situation, $\boldsymbol{i^t}$ is the situation $\boldsymbol{i}$ and $\boldsymbol{o^i}$ is the outcome of situation $\boldsymbol{i}$. The obtained function $\boldsymbol{\mu}$ is modified to a monotone function $\boldsymbol{\pi}$ corresponding to some decision. This function is aggregated with a utility function $\boldsymbol{u}$ in order to determine optimal decision in a similar way as described in the section about possibilistic decision theory.

Negotiation may be quite expensive and time consuming, especially in scenarios with a large number of agents. In such situations it is important to determine with whom there is a higher chance of successful negotiation and reaching better agreements. The possibility-based mechanism can predict the ordering of potential partners by placing the most prospective partners for negotiation at the top and the less prospective further in the ordering. This allows choosing from the whole set of all the agents a subset of the most prospective ones for negotiation.

Brzostowski and Kowalczyk (2005) presented a scenario in which reasoning is done by the main contractor who is offered services from a number of agents. The contractor may use them individually or aggregate them as a compound service. From the set of agents representing services we need to choose the subset of most prospective agents. In order to do this we model the system of all potential partners using the tools of possibility theory.

We noted previously that the obtained possibility distribution describes the likelihood of successful negotiation and is derived from the history of previous interactions. The distribution encodes the prediction of the main contractor about preferences of his negotiation partners. Based on this function and the utility of the main contractor the estimation of the outcome of current negotiation can be

[7] An example of T-norm that was used here is the min operator.

calculated. In order to do so the calculation of the qualitative expected utility is required; it is obtained by the aggregation of possibility distribution and the main contractor's utility function. The estimation of negotiation outcome allows us to rank the negotiation partners. The final ordering gives the information with whom to negotiate first and with whom to negotiate later.

The mechanisms described above treats the uncertainty about attributes or negotiation outcomes and are suitable in situations when prediction is required because the negotiation partner does not want to reveal his private information. However, information revelation occurs in some real world problems. Such problems may be: meeting scheduling, planning or resource allocation. In such scenarios the agent does not need to learn because the knowledge about partners' preferences is given, although it may sometimes contain some uncertainties. For solving such negotiation encounters the constraint based reasoning may be used.

15.5.7 Q-learning

The multiagent system SMACE (Oliveira and Rocha 2000) combines the idea of decision functions and reinforcement learning algorithms into a new approach called Q-learning. An agent that uses reinforcement learning takes actions in a dynamic environment and is rewarded or punished depending on the consequences of actions taken. Learning agents have to explore the environment by performing actions. An agent receives feedback from the reward function and based on this feedback, learns which actions should be carried out in which states. Q-learning is an example of reinforcement learning based on the update of Q values.

Faratin *et al.* (1998) defined the agent's current action (counterproposal made by agent a to agent b at time t) in p-issue and m-tactics negotiation as the matrix of weights:

$$\Gamma_{a \to b}^{t} = \begin{pmatrix} \omega_{11} & \omega_{12} & \cdots & \omega_{1m} \\ \omega_{21} & \omega_{22} & \cdots & \omega_{2m} \\ . & & & \\ . & & & \\ . & & & \\ \omega_{p1} & \omega_{p2} & \cdots & \omega_{pm} \end{pmatrix}$$

where ω_{ij} is a weight corresponding to a negotiation issue i and tactic j. The strategy in their approach is a function f mapping the action $\Gamma_{a \to b}^{t_n}$ in time t_n and the agent's mental state $MS_a^{t_n}$ in time t_n to the new action $\Gamma_{a \to b}^{t_{n+1}}$ for time t_{n+1}:

$$\Gamma_{a \to b}^{t_{n+1}} = f(\Gamma_{a \to b}^{t_n}, MS_a^{t_n})$$

But in such a model the question of specifying function f remains open. The Q-learning may be regarded as a complement to this model because it allows learning and updating of the so-called ***Q*** values. The ***Q*** values are some kind of rewards or utilities assigned to each pair of action and state ***Q***(***i***,***a***). At first, the optimal action has to be determined by using ***Q*** values acquired so far. The chosen action should maximize the expected utility. After determining an appropriate action the ***Q*** value can be updated. The whole process may be described by the following formula:

$$Q(i,a) = Q(i,a) + \alpha[r(i) + \gamma \max_{a'} Q(j,a') - Q(i,a)]$$

where ***α*** is a learning rate, ***r(i)*** is a reward gained by performing action ***a*** in state ***i***, γ is a discount parameter, ***j*** is the state attained. The reward may be positive or negative depending on whether the action gives good or bad results. The results are the achieved deals and their utilities.

The state in the negotiation scenario may be described by such factors as the number of negotiating agents and time left for negotiation. The action is encoded by the sequence of weights corresponding to the applied tactics.

An agent applying the mechanism described above is able to improve its performance by using experience to learn what tactics should be employed in what situations. However, the main disadvantage of this approach is that the knowledge acquisition process requires many trials. Q-learning also requires the determination of balance between trying new actions and applying the old ones that already proved to be good. For more details on Q-learning see Russel and Norvig (2003).

15.5.8 Evolutionary Computing

Another trial-and-error approach for learning good strategies is evolutionary computing. Evolutionary algorithms enable searching the space of potential solutions by applying the principle of natural law stating that fit parents would most likely produce fit children in the process of reproduction. The candidate solutions are called chromosomes. The search starts by creating a first random population of chromosomes chosen from the space of potential solutions. The next generation is created in two steps. During the first step which is the recombination, the chromosomes from the previous generation are paired two-by-two and “crossed over”. The second step is mutation - the change of some part of the chromosome. This operation models errors occurring while copying genes from the previous generation.

The object to be encoded as chromosome is the agent’s strategy. The first paper applying evolutionary computation for negotiation automation (Oliver 1997) had been published before the idea of decision-functions-based strategies was proposed. Therefore, the notion of strategy in this paper is defined in a much simpler way as a threshold decision rule. An agent applying this rule accepts an offer in the first step, which exceeds some threshold T_1. If the threshold is not exceeded, the agent makes a counterproposal. If the opponent does not agree on this proposal it makes a subsequent proposal. Our agent accepts this proposal if it exceeds the next threshold T_2. Again, if the opponent does not agree it makes a counterproposal and the process continues until an agreement is reached or one of the sides stops the negotiation. The strategy defined in this manner is encoded as a sequence of thresholds and counterproposals.

The learning process is done in the following way. For both negotiating agents (we consider bilateral negotiation here) the random population of candidate solutions is generated. Both sides select strategies from their populations which are then tested in the negotiation process. After negotiation, agents assign fitness to the tested strategies according to their performance during the encounter. The selection and test is carried out a number of times close to the number of strategies in the population, so that each strategy is chosen approximately once. Having the fitness assigned to population members the new population is created using a genetic algorithm and the process of testing is done again. The higher the fitness the higher the chance that a strategy will be chosen for reproduction. The whole process of population update and testing is repeated until an exit condition is satisfied.

The other papers dealing with the evolutionary computation approach for negotiation usually apply similar mechanisms of learning to that described above but the notion of strategy is more complex. Matos *et al.* (1998) encodes in the chromosome information like deadlines, domains of each attribute, monotonicity of each attribute, weights of all tactics and parameters specifying each tactic. Some reproduction mechanisms may be more sophisticated. Gerding *et al.* (2004) use the same notion of strategy as Olivier that is the sequence of thresholds and counter-proposals. The main difference is the application of mutation as a reproduction operator in this case. The recombination is not used because the authors claim that it does not have a large influence on evolving system.

15.6 Applications of Negotiation Software Agents

Agents can be applied to a variety of negotiation problems in e-commerce, planning, resource allocation, scheduling, and so on. Auctions, in addition to negotiations, have been recently widely used in resolving these these problems. In general, auctions are considered as “a market institution with an explicit set of rules determining resource allocation and prices on the basis of bids from the market participants” (McAfee and McMillan 1987). Although traditionally auctions have been considered distinct from negotiations, recent changes in auction mechanisms allowed their use for resolving more types and domains of problems (Ströbel and Weinhardt 2003).

One of the challenges in designing a negotiation software agent for one-sided auctions is the ability to join multiple auctions. By participating in many auctions the agent can purchase the required number of goods for the low price. Preist *et al.* (2000, 2001) describe agents that enable the identification of the most beneficial auctions (closing with low price) and the coordination of bidding in these auctions in order to win the lowest possible price. In this approach an appropriate coordination algorithm allowing the purchasing of the right number of goods is needed. The proposed learning mechanism allows for the construction of the belief function in which the probability that some number of participants value the good with valuation higher than some specific value is included. Based on this belief function an agent can decide whether to bid higher in the terminating auction or to place bids in the remaining auctions. This is done using a comparison of calculated expected utilities for these auctions.

Apart from monitoring auctions and selecting the ones to participate in, the decision making concerning how to bid remains a difficult problem. This problem is addressed by Anthony *et al.* (2001, 2003). The authors propose the design of bidding strategies based on decision functions (Faratin 2000). The marketplace is simulated with various types of one sided auctions. The current maximum bid is determined considering the bidding constraints such as: time left, remaining auctions left, the participant's desire to bargain and participant's level of desperateness. Based on the value of the current maximum bid potential, auctions are selected and the bid for each of these auctions is calculated. Then the auction and corresponding bid with the highest expected utility is selected. The authors also search the set of bidding strategies offline, using genetic algorithms in order to determine the best strategies.

Byde *et al.* (2002) developed a sophisticated decision theoretic framework that enables agents to bid rationally across multiple auctions. The framework is described for a few types of auctions. The rational agent will bid in an auction if the expected future utility of bidding exceeds the expected future utility of not bidding. To make this decision, agents need to estimate the future utility first. As a solution to this the authors propose two alternatives: backward induction or fixed auction strategies. Finally a heuristic algorithm allowing an agent to make appropriate decisions is described.

One interesting approach is described by Garcia *et al.* (1998) in which possibility-based decision theory is employed to calculate the best bid. The uncertainty about opponent's behavior is modeled by a possibility distribution that is obtained by case-based reasoning. This possibility distribution enables making decisions about the bidding strategy.

The traditional setting of auctions may be extended by introducing new attributes other than just the price of the goods under consideration. The multiattribute English auction is considered by Dawid *et al.* (2003). A scenario with one buyer and multiple sellers is presented. The situation here is more complex than in a single-attribute auction. Due to the multidimensionality, the utility function for each buyer and seller has to be specified. The seller specifies his requirements by announcing the scoring function, the minimum increment and the maximum number of rounds. Two types of auctions are described: sequential and simultaneous. The seller's bidding strategy is determined as an action maximizing the expected utility. In the continuous double auction both sellers and bidders submit their proposals and the process is stopped when the offer of one party meets the offer of another one. There are various types of strategies applied in this kind of auction. The continuous double auction is more efficient and flexible than the one-sided auctions. The mechanisms deciding what bid to make vary from very simple to very sophisticated.

One of the first approaches to bidding involved agents that used a "zero intelligence" strategy (Gode and Sunder 1993). This strategy generated a random bid within the allowed range. The "zero intelligence" strategy turned out to be quite efficient when compared with other, more intelligent strategies. Subsequently, other complex decision-making strategies have been proposed. Park *et al.* (1999) proposed an adaptive agent bidding strategy based on stochastic modeling. The authors claim that stochastic modeling is a good substitute for full rationality, but because of the computational costs and time consumption it should be decided in what situations to use it. Therefore the agent should be adaptive in the sense that it uses the appropriate mechanism when it is necessary.

Dynamic programming was also applied in auctions (Tesauro and Bredin 2002). They developed an algorithm for both sides: buyer and seller participating in CDA, based on fuzzy logic. They used heuristic fuzzy rules and fuzzy reasoning to calculate the optimal bid given the current state of the market. The agent based on this approach can also adapt its bidding behavior to changes occurring in the environment.

In the following section, we will give an overview of several applications that enable intelligent agents to negotiate and take part in electronic auctions. AuctionBot (Wurman *et al.* 1998) was a project at the University of Michigan to develop a flexible, scalable, and robust online auction server for many auction types. AuctionBot provides the service of hosting and processing an auction according to user's preferences via a Web interface (for human users) and an Application Programming Interface (for software agents). AuctionBot supports the widest possible range of different auction types by decomposing the auction design space to different parameters (Wurman *et al.* 2001), for example the number of buyers and sellers to participate in the auction, closing conditions, and the allocation policy. Users can create new auctions by specifying those parameters. Buyers and sellers can bid according to the auction rules using the Web interface or by allowing the software agents to bid on behalf of a user using the programming interface. AuctionBot was used to host the first two Trading Agent Competitions in 2000 and 2001.

Kasbah (Maes 1998) is an online virtual marketplace where users can create agents to buy and sell goods on their behalf. Kasbah is a multi-agent system, in which agents must act and communicate according to a specified protocol. When a user creates a selling agent, he gives a description of the item to sell. In addition, the user must specify parameters on a very high level of abstraction, such as the desired date to have the item sold, the desired price, and the lowest acceptable price. In addition, the user has control over the agent's negotiation strategy, that is, the user can specify the decay function (linear, quadratic, cubic) in order to lower the price and time. The agent can be specified to ask its owner before finalizing a deal. All these parameters can be changed by the user at any time after the agent has been created. The definition of a buyer agent works analogously.

The selling agent then proactively searches for other agents that are interested in purchasing this item and starts the negotiation process, which works straightforwardly. After a selling agent has found a buyer agent interested in the offered item, buyer agents are allowed to offer a bid to selling agents without any further restrictions regarding price, time, *etc.* Selling agents only reply with either "yes" or "no". Once a buying agent and a selling agent have reached an agreement on a specific price, both users are asked for their respective approval.

Given this protocol, Kasbah users can actually select from three different buying and selling strategies, respectively. If the user selects for example a linear increasing function for a buying agent, he or she follows an anxious negotiation strategy. This is due to the fact that the user must increase offers quickly in order to be able to win the negotiation. According to Maes (1998) the simplicity of this negotiation process is necessary for user understanding and user acceptance.

Whereas in Kasbah the only negotiable attribute is price, MIT's Tête-à-Tête project (Maes and Guttman 1999) provides buying and selling agents to cooperatively negotiate across multiple terms of a transaction, for example, warranty, delivery time, return policy, and other merchant value-added services. Based on bilateral argumentation, Tête-à-Tête's negotiation process works on XML documents that describe proposals, critiques, and counterproposals, thus making the negotiation process more complex than Kasbah's one. A buying agent receives proposals from multiple selling agents and evaluates them according to the user's multiattribute utility functions. If the agent is not satisfied with proposals, it can critique them using one or many attributes and broadcast this to the selling agents. After receiving critiques, selling agents can then use them to create better counterproposals. Using critiques, selling agents can place constraints on product features in order to influence the decision of whom to buy from and what to buy.

A slightly different negotiation protocol is implemented by Magnet (Collins *et al.* 1998) in which agents are used to negotiate contracts and later monitor their execution. First, a customer (buying agent) publishes a Request for Quotes. Then, suppliers (selling agents) respond by providing an offer detailing the price of the requested resource over a specified time period. Customers evaluate bids looking at price, risk, and time constraints and finally select the optimal set of bids that satisfy their goals. Suppliers are notified about the result. Second, an execution manager component is initiated to monitor the fulfillment of the contract and start re-negotiation process if necessary.

e-Negotiation agents (eNAs) and fuzzy e-negotiation agents (FeNAs) (Kowalczyk 2002) are prototypical intelligent trading agents that autonomously negotiate multiple terms of transactions in e-commerce trading. These agents engage in integrative negotiations in the presence of limited common knowledge about other agents' preferences, constraints and objections through an iterative exchange of multiattribute offers and counteroffers. Fuzzy eNAs allow the specification of fuzzy constraints and preferences. The FeNAs environment consists of many autonomous trading agents representing buyers and sellers that can engage in concurrent bilateral negotiations according to a number of user-selected negotiation strategies. The eNAs and FeNAs agents have been demonstrated with a number of test-beds of e-commerce trading (Kowalczyk and Bui 2003).

Kersten and Noronha (1999) proposed negotiation software agents that provide information and knowledge (*e. g.* statistics and inferences) about past negotiations, scan the negotiation transcripts and other process descriptions, and then provide a comparison of situations, interests and issues of past problems to the current problem. These agents may also receive knowledge from various sources, such as other agents, the environment, user input and databases, then interpret and understand that knowledge and intelligently use information to assist the negotiator throughout the negotiation processes (Torsun 1995).

Kersten and Lo (2003) developed Aspire, a Web-based system comprising software agents, and negotiation and decision support systems. Aspire's functionalities include supporting negotiators, providing context-dependent advice, and undertaking certain activities autonomously. A software agent monitors the process, facilitates the use of the Web-based negotiation support system, interprets the negotiators' activities and provides methodological advice. The architecture of the system separates user support functions from the autonomous software activities, separation of the support for individuals from facilitation and mediation, scalability and the ability to provide linkages with the existing software.

eAgora (Chen *et al.* 2004) is an e-marketplace that allows buyers and sellers to engage in multi-issue negotiations. Its services include a software agent that generates and critiques offers. A usability test for comparing negotiations with and without the agent, found the agent's services were useful and partial negotiation automation is desired.

For projects focusing on mobile networks, in which for example mobile agents are used as user representatives in online auctions, we refer to Agora (Fonseca *et al.* 2001), Impulse (Youll *et al.* 2000), MAgNET (Dasgupta 2002), and BiddingBot (Fukuta *et al.* 2001).

15.7 Conclusion

In this chapter, we discussed research on e-negotiation systems and presented theory and applications of software agents for electronic negotiations. Types of negotiation agents and their roles and requirements were discussed and various models for negotiation software agents were reviewed. We also presented applications of negotiation software agents.

We conclude this chapter by emphasizing that software agent technologies should be regarded as tools for effective support of negotiations. The research questions should focus on the problems of the user (or principal), not the capability of the agents and availability and feasibility of technologies. Negotiations are in many cases ill-structured problems that require human ability to reformulate the issues, redefine the negotiation process, understand participant's interests, and develop strategies and tactics. However, most research on the use of software agents in negotiations has focused on automation of the communication and decisionmakings in the negotiation process. This approach can only fit into negotiation processes involving well-structured problems where human learning and socialization attempts to build business or other relationships do not have significant effects.

DSSs and NSSs have focused on ill-structured negotiation problems that take shape and have issues clarified during the negotiation through human intervention as well as well-structured problems. The negotiation software agents have advantages in automating well-structured problems. From our point of view, negotiation software agents may take over well-defined and structured activities in a negotiation but it is not necessary for agents to handle all the tasks. For example, the agent may present offers, seek information about similar negotiation situations, collect information about the counterparts, and alert the principal if predefined conditions are violated. The ill-defined and ambiguous issues, decisions regarding the relationship between the parties, modification of the rules and parameters are better left to the principals. Therefore, we believe that it is important to first consider the effective mixture of both autonomous agents and DSS/NSS.

In this chapter, we approached software agents and e-negotiation systems from the perspective of the hybrid NSA/DSS/NSS architecture that allows for human-system-agent interactions. Such an integrated architecture allows utilizing the strengths and capabilities of the methods and models that are embedded in the support systems and software agents. It also allows us to better define the roles of the individual components, the collaboration patterns, and the scope and levels of the agents' autonomy and the systems support.

Acknowledgements

This work has been partially supported by the grants from the Department of Education, Science and Training of Australia (AU-DEST-CG060081), the Social Sciences and Humanities Research Council of Canada (SSHRC), and the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

Anthony P, Hall W, *et al* (2001) Autonomous Agents for Participating in Multiple Online Auctions. Proceedings of the IJCAI Workshop on E-Business and Intelligent Web, ACM

Anthony P, Jennings NR (2003) Developing A Bidding Agent for Multiple Heterogeneous Auctions. ACM Transactions on Internet Technology 3(3): 185-217

Binmore K (1992) Fun and Games, a Text on Game Theory, D. C. Heath and Company.

Binmore K, Vulkan N (1999) Applying Game Theory to Automated Negotiation. Netnomics 1(1): 1-9

Brzostowski J, Kowalczyk R (2005) On Possibilistic Case-based Reasoning for Selecting Partners in Multi-Agent Negotiation. Proceedings of The 17th Joint Australian Conference on Artificial Intelligence, Cairns, Australia, Springer (in print).

Bui T, Yen J, et al (2001) A Multi-Attribute Negotiation Support System with Market Signaling for Electronic Markets. Group Decision and Negotiation 10(6): 515-537

Byde A, Preist C, *et al.* (2002) Decision Procedures for Multiple Auctions. Proceedings of the first international joint conference on Autonomous agents and multi-agent systems: Part 2, ACM, New York

Caglayan A, Harrison C (1997) Agent Sourcebook. New York, John Wiley & Sons, Inc.

Chen E, Kersten GE, et al (2004) Agent-supported Negotiations on E-marketplace. International Journal of Electronic Business: (forthcoming).

Collins J, Tsvetovat M, *et al.* (1998) MAGNET: A Multi-Agent Contracting System for Plan Execution. Workshop on Artificial Intelligence and Manufacturing: State of the Art and State of Practice, Albuquerque, USA, AAAI Press

Cybersettle (2000) White Plains, Cybersettle.com. 2004

Dasgupta P. (2002) Fault tolerance in MAgNET: A Mobile Agent E-commerce System. Proceedings of the Sixth International Conference on Internet Computing, Las Vegas, USA

David E, Azoulay-Schwartz R, *et al.* (2003) Bidders' Strategy for Multi-Attribute Sequential English Auction with a Deadline. Proceedings of the second international joint conference on Autonomous agents and multi-agent systems, Melbourne, Australia, ACM

Dubois D, Prade H (1996) Combining hypothetical reasoning and plausible inference in possibilistic logic. Multiple Valued Logic 1: 219-239

Dubois D, Esteva F, et al (1998) Fuzzy set modelling in case-based reasoning. International Journal of Intelligent Systems 13: 345-373

Ehtamo H, Kettunen E, et al (2001) Searching for Joint Gains in Multi-party Negotiations. European Journal of Operational Research 130: 54-69

Faratin P (2000) Automated Service Negotiation Between Autonomous Computational Agents. London, U.K., Department of Electronic Engineering, Queen Mary & Westfield College

Faratin P, Sierra C, *et al.* (1998) Negotiation Decision Functions for Autonomous Agents. Robotics and Autonomous Systems 24(3-4): 159-182.

Fatima S, Wooldridge MJ, et al (2002) The Influence of Information on Negotiation Equilibrium. Agent-Mediated Electronic Commerce IV, Designing mechanisms and systems, Springer. 2531: 180-193.

Fisher R, Kopelman E, et al (1994) Beyond Machiavelli. Tools for Coping with Conflict. Cambridge, MA, Harvard University Press.

Fisher R, Ury W, et al (1991) Getting to Yes: Negotiating Agreement Without Giving In. New York, Penguin Books.

Franklin S, Graesser A (1997) Is it an Agent, or Just a Program? A Taxonomy for Autonomous Agents. Intelligent Agents III: Agent Theories, Architectures, and Languages. J. P. Muller, M. J. Wooldridge and N. R. Jennings. Berlin, Springer Verlag: 21-36.

Fukuta N, Ito T, et al (2001) A Framework for Cooperative Mobile Agents and its Case Study on Bidding Bot. JSAI 2001 International Workshop on Agent Based Approaches in Economic and Social Complex Systems.

Garcia P, Gimenez E, et al (1998) Possibilistic-based Design of Bidding Strategies in Electronic Auctions. Proceedings of the 13th European Conference on Artificial Intelligence, Brighton, UK, Wiley

Gerding EH, Somefun DJA, *et al.* (2004) Multi-Attribute Bilateral Bargaining in One-to-Many Setting. Amsterdam, Centre for Mathematics and Computer Science (CWI)

Gode D, Sunder S (1993) Lower Bounds for Efficiency of Surplus Extraction in Double Auctions. The Double Auction Market: Institutions, Theories, and Evidence. D. Friedman and J. Rust. Reading, MA, Addison-Wesley: 199-219

Gulliver PH (1979) Disputes and Negotiations: A Cross-Cultural Perspective. Orlando, FL, Academic Press

Harsanyi JC (1997) Rational Behaviour and Bargaining Equilibrium in Games and Social Situations. Cambridge, Cambridge University Press

Hiltz SR, Turoff M (1978) The Network Nation: Human Communication via Computer. Reading, MA: Addison-Wesley.

Inspire (1996). Inspire ENS. Ottawa, G. Kersten & InterNeg. 2004.

Jarke M, Jelassi MT, et al (1987) MEDIATOR: Toward a Negotiation Support System. European Journal of Operational Research 31(3): 314-334.

Jelassi MT, Foroughi A (1989). Negotiation Support Systems: An Overview of Design Issues and Existing Software. Decision Support Systems 5: 167-181.

Jennings NR, Wooldridge NM (1998) Applications of Intelligent Agents. In: N. Jennings and M. Wooldridge (Eds) Agent Technology: Foundations, Applications and Markets, Springer-Verlag

Jennings NR (2000) On agent-based software engineering. Artificial Intelligence 117(2): 277-296

Jennings NR (2001). An Agent-Based Approach for Building Complex Software Systems. Communications of the ACM 44(4): 35-41

Jennings NR, Faratin P, et al (2001). Automated Negotiations: Prospects, Methods and Challenges. Group Decision and Negotiation 10(2): 199-215
JointGains (2000). Joint Gains Negotiation Support in the Internet. Helsinki, System Analysis Laboratory. 2004
Keen PGW, Scott-Morton MS (1978) Decision Support Systems: An Organizational Perspective. Reading, Addison-Wesley
Kersten GE (1993) Rule-based Modeling of Negotiation Processes. Theory and Decisions 34(2)
Kersten GE (1997) Support for group decisions and negotiations. In: J. Climaco (Ed.) An Overview, in Multiple Criteria Decision Making and Support, Heidelberg: Springer Verlag
Kersten GE (1999) Decision-making and Decision Support. Decision Support Systems for Sustainable Development in Developing Countries. G. E. Kersten, Z. Mikolajuk and A. Yeh. Boston, Kluwer: 29-52
Kersten GE (2001) Modeling Distributive and Integrative Negotiations. Review and Revised Characterization. Group Decision and Negotiation 10(6): 493-514
Kersten GE, Lo G (2003) Aspire: Integration of Negotiation Support System and Software Agents for E-Business Negotiation. International Journal of Internet and Enterprise Management 1(3): 293-315
Kersten GE, Noronha S (1999) WWW-based Negotiation Support: Design, Implementation, and Use. Decision Support Systems 25: 135-154
Kersten GE (2004) E-negotiation Systems: Interaction of People and Technologies to Resolve Conflicts, UNESCAP Third Annual Forum on Online Dispute Resolution, Melbourne, Australia
Kim J, Segev A (2003). A Framework for Dynamic eBusiness Negotiation Processes. IEEE Conference on E-Commerce, New Port Beach, CA
Kinny D, Georgeff M, et al (1996) A Methodology and Modelling Technique for Systems of BDI Agents, Australian Artificial Intelligence Institute.
Korhonen P, Moskowitz H, *et al.* (1986) An Interactive Approach to Multiple Criteria Optimization with Multiple Decision-Makers. Naval Research Logistics Quarterly 33: 589-602.
Kowalczyk R (2000) On negotiation as a distributed fuzzy constraint satisfaction problem. Proceedings of the Third International Symposium on Soft Computing for Industry of the World Automation Congress
Kowalczyk R (2002) Fuzzy eNegotiation Agents. Journal of Soft Computing, Special Issue on Fuzzy Logic and the Internet 6(5): 337-347
Kowalczyk R, Bui V (2003) On Constraint based Reasoning in eNegotiation Agents. In: F. Dignum and U. Cortes (Eds) Agent Mediated Electronic Commerce III. Springer Verlag, pp 31-46
Kraus S (2001) Strategic Negotiation in Multiagent Environments. Cambridge, MIT Press.
Lempereur A (2004) Innovation in Teaching Negotiation. Toward A Relevant Use Of Multimedia Tools. International Negotiations Journal 9(1): 141-160.
Lewicki RJ, Saunders DM, *et al.* (1999) Negotiation. Boston, MA, McGraw-Hill.
Luck M, McBurney P, *et al.* (2003) Agent Technology: Enabling Next Generation Computing. A Roadmap for Agent-Based Computing. AgentLink II. http://www.agentlink.org/roadmap/distribution.html
Luo X, Jennings NR, et al (2003) A Fuzzy constraint based model for bilateral multi-issue negotiation in semi-competitive environments. Artificial Intelligence 148(1-2): 53-102.
Luo X, Lee JH, et al (2003) Prioritised Fuzzy Constraint Satisfaction Problems: Axioms, Instantiation and Validation. Fuzzy Sets and Systems 136(2): 155-188.
Maes P (1998) Artificial Life Meets Entertainment: Life-like Autonomous Agents. Communications of the ACM 38(11): 108-114

Maes P, Guttman RH (1999) Agents that can buy and sell. Communication of the ACM 42(3): 81-91

Matos N, Sierra C, et al (1998) Determining Successful Negotiation Strategies: An Evolutionary Approach. Proceedings of the 3rd International Conference on Multi-Agent Systems, Paris, France

Matwin S, Szpakowicz S, et al (1987) Logic-Based System for Negotiation Support. Proceedings of the 1987 Symposium on Logic Programming, San Francisco, IEEE Computer Society Press

McAfee RP, McMillan J (1987) Auction and bidding. Journal of Economic Literature 25: 699-738

Moore D, Kurtzberg T, et al (1999) Long and Short Routes to Success in Electronically Mediated Negotiations: Group Affiliations and Good Vibrations. Organizational Behavior and Human Decision Processes 77(1): 22-43

Neumann D, Benyoucef M, et al (2003) Applying the Montreal Taxonomy to State of the Art E-Negotiation Systems. Group Decision and Negotiation 12(4): 287-310.

NovaForum (2000). The Electronic Courthouse. Ottawa. 2004

Nwana HS, Ndumu DT (1998) A Brief Introduction to Software Agent Technology. In: N. Jennings and M. Wooldridge (Eds) Agent Technology: Foundations, Applications & Markets, Springer Verlag: 29-47

Oliveira E, Rocha AP (2000). Agents advances features for negotiation in electronic commerce and virtual organisations formation process. In: C. Sierra and F. Dignum (Eds) European Perspectives on Agent Mediated Electronic Commerce, Springer. 1991

Oliver JR (1997) A Machine-Learning Approach to Automated Negotiation and Prospects for Electronic Commerce. Journal of Management Information Systems 13(3): 83-112

Park S, Durfee H, et al (1999) An Adaptive Agent Bidding Strategy based on Stochastic Modelling. Proceedings of the third annual conference on Autonomous Agents, ACM, New York

Preist C (2000) Algorithm Design for Agents Which Participate in Multiple Simultaneous Auctions. Hewlett-Packard, Bristol

Preist C, Byde A, et al (2001) Economic Dynamics of Agents in Multiple Auctions. Proceedings of the fifth international conference on Autonomous agents, ACM, New York

Purdy JM, Neye P (2000) The Impact of Communication Media on Negotiation Outcomes. The International Journal of Conflict Management 11(2): 162-187

Raiffa H, Richardson J, et al (2003) Negotiation Analysis. The Science and Art of Collaborative Decision-making. Cambridge, Harvard University Press

Rangaswamy A, Eliasberg J, et al (1989) Developing Marketing Expert Systems: An Application to International Negotiations. Journal of Marketing 53: 24-39

Rangaswamy A, Shell GR (1997) Using Computers to Realize Joint Gains in Negotiations: Toward an Electronic Bargaining Table. Management Science 43(8): 1147-1163

Rebstock M (2001). An Application Architecture for Supporting Interactive Bilateral Electronic Negotiations. Electronic Commerce and Web Technologies. K. Bauknecht, S. K. Madria and G. Pernul. Berlin, Springer: 196-205

Rosenschein, JS, Zlotkin G (1994) Rules of Encounter. Cambridge, The MIT Press

Roth AE (1995) Bargaining: Economic Theories of Bargaining. Social Science Encyclopedia. A. Kuper. London, Routledge.

Rubinstein, A (1998). Modelling Bounded Rationality. Cambridge, USA, The MIT Press.

Russell, S. and P. Norvig (2003). Artificial Intelligence: A Modern Approach, Prentice Hall.

Sathi A, Fox MS (1989) Constraint-directed negotiation of resource reallocation. In: L. Gasser and M. Huhns (Eds.), Distributed Artificial Intelligence Volume II, pp 163-195, San Mateo, California, 1989. Morgan Kaufmann.

Schoop M, Quix C (2001) DOC.COM: A Framework for Effective Negotiation Support in Electronic Marketplaces. Computer Networks 37(2): 153-170.
Simon HA (1977) The New Science of Management Decision. Englewood Cliffs, N.J., Prentice-Hall.
Sprague RM, Carlson ED (1982) Building Effective Decision Support Systems. Englewood Cliffs, Prentice-Hall
Stahl I. (1972) Bargaining Theory. Stockholm, Stockholm School of Economics.
Starke K, Rangswamy A (1999) Computer-Mediated Negotiations: Review and Research Opportunities. University Park, eBusiness Research Center: 37
Sycara, DZ (1997). How can an agent learn to negotiate. In: M. W. J. Mueller and N. Jennings (Eds), Intelligent Agents III. Agent Theories, Architectures, and Languages, Springer Verlag
Tesauro G, Bredin JL (2002) Strategic Sequential Bidding in Auctions using Dynamic Programming. Proceedings of the first international joint conference on Autonomous agents and multi-agent systems: part 2, ACM, New York
Thompson L, Nadler J (2002) Negotiating via Information Technology: Theory and Application. Journal of Social Studies 58(1): 109-124
Torsun IS (1995).Foundations of Intelligent Knowledge-Based Systems. Bradford, Academic Press Inc.
Ury W (1993) Getting Past No. Negotiating your Way from Confrontation to Cooperation. New York, Bantam Books
Vahidov R, Fazlollahi B (2004) Pluralistic Multi-Agent Decision Support System: a Framework and an Empirical Test. Information & Management 41(7): 883-898
Von Neumann J, Morgenstern O (1944) Theory of Games and Economic Behavior. Princeton: Princeton University Press
Ströbel M, Weinhardt C (2003) The Montreal Taxonomy for Electronic Negotiations. Group Decision and Negotiation, 12(2): pp. 143-164
Wetlaufer GB (1996) The Limits of Integrative Bargaining. Georgetown Law J. 85: 1-25
Wolfstetter E (1996) Auctions: An Introduction. Journal of Economic Surveys 10(4): 367-420
Wooldridge M (1999) Intelligent Agents. Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. G. Weiss. Cambridge, The MIT Press: 27-78.
Wooldridge M, Jennings NR (1995) Intelligent Agents: Theory and Practice. Knowledge Engineering Review, 10(2): 115-152
Wurman PR, Wellman MP, et al (1998) The Michigan Internet AuctionBot: A configurable auction server for human and software agents. Second International Conference on Autonomous Agents, Minneapolis, MN, pp 301-308.
Wurman PR, Wellman MP, *et al.* (2001) A Parametrization of the Auction Design Space. Games and Economic Behavior 35: 304-338.
Yokoo M, Durfee EH et al (1998) The distributed constraint satisfaction problem: formalization and algorithms, IEEE Transactions on Knowledge and Data Engineering, 10(5):673-685, 1998
Youll J, Morris J, *et al.* (2000). Impulse: Location-based Agent Assistance. Proceedings of the Fourth International Conference on Autonomous Agents (Agents 2000), Barcelona, Spain.
Young OR (1975) Bargaining. Formal Theories of Negotiations. University of Illinois Press, Urbana, IL
Zeng D, Sycara K (1997) Benefits of Learning in Negotiation. Proceedings of AAAI-97
Zeng D, Sycara K (1998) Bayesian learning in negotiation. Int. J. Human-Computer Studies 48: 125-141
Zlotkin G (1996) Mechanisms for Automated Negotiation in State Oriented Domains Journal of Artificial Intelligence Research 5: 163 - 238

16

Knowledge-intensive Collaborative Decision Support for Design Process

Xuan F Zha and Ram D Sriram

Manufacturing System Integration Division, National Institute of Standards and Technology, Gaithersburg, MD 20899, USA e-mail: zha@cme.nist.gov

In this chapter, we describe a hybrid decision model and a multiagent framework for collaborative decision support in the design process. The proposed knowledge-based collaborative decision support model can quantitatively incorporate qualitative design knowledge and preferences for multiple, conflicting attributes stored in a knowledge repository so that a better understanding of the consequences of design decisions can be achieved from an overall perspective. The multiagent framework provides an efficient decision support environment involving distributed resources to shorten the realization of products with optimal life-cycle performance and competitiveness. The developed model and framework are generic and flexible enough to be used in a variety of design decision problems. The framework is illustrated with an application in concept evaluation and selection in power-supply product family design for mass customization.

16.1 Introduction

Engineering design is essentially a collaborative decision-making process that requires rigorous evaluation, comparison and selection of design alternatives as well as optimization from a global perspective on the basis of different classes of design criteria. Increasing design knowledge and supporting designers to make correct and intelligent decisions can increase design efficiency. Thus, a design strategy must be devised to specifically address all aspects of design including process modeling, knowledge modeling, decision support, and the inherent complexity arising from representing physical design problems using idealized computer-based models. Such a strategy can, then, lead to the identification and development of knowledge decision support techniques that play a critical role in enabling designers to make intelligent decisions towards improving the overall quality of the products designed.

This chapter aims to develop a knowledge-supported decision support methodology for the smooth integration of stakeholders involved in collaborative product development and improved product performance. The goal is to develop a sound, robust, practical trade-off-based design decision model that can quantitatively incorporate qualitative knowledge and preferences for multiple, conflicting attributes stored in a knowledge repository. The focus in this chapter is to establish a knowledge-based decision model and framework for collaborative design.

The organization of this chapter is as follows. Section 16.2 reviews the previous research related to design decision support and current status. Section 16.3 discusses the design decision support process and decision-based design. Knowledge-intensive decision support for design process is highlighted. Section 16.4 proposes a knowledge-based decision model. Section 16.5 discusses collaborative decision-making mechanisms. Section 16.6 proposes a multiagent collaborative decision support framework. Section 16.7 provides the application of the proposed model in concept evaluation and selection. Section 16.8 provides a case study. Section 16.9 summarizes the chapter and points out opportunities for future work.

16.2 Literature Review

Design decision support problems necessitate the search for superior or satisfying design solutions (Simon 1976), especially in the early stages of design, when the all of the information needed to model a system comprehensively may not be available. Current research in design decision support (particularly pertaining to decision-based design) is focused on enabling technologies to assist product designers to make decisions in the design process (Rosen *et al.* 2000, Mistree *et al.* 1995), where primary emphasis is on support for information management related to decision making. Generally, the literature on design evaluation and selection decision support can be classified into six categories (Jiao and Tseng 1998): 1) multi-criteria utility analysis, 2) fuzzy set analysis, 3) probability analysis, 4) the hybrid approach, 5) design analytic methodology, and 6) the information content approach (Suh 1990).

With the emergence of collaborative design, researchers are addressing enabling technologies or infrastructure to assist product designers in the computer or network-centric design environment (Sriram 2002, Rosen *et al.* 2000). Some recent techniques are intended to help designers collaborate or coordinate by sharing product information and manufacturing services through formal or informal interactions, while others are geared towards conflict management. Most decision support programs can only calculate satisfaction levels. There is a need for adding unique analysis and reporting features, including: probability that a particular alternative is the best choice; assessment of the level of consensus for each alternative; guidance on what should be done next; and documentation of the entire decision-making process. In the early stages design decisions are ill-structured and often supported with scarce information. Multiple potential solutions and limited predictability all contribute to the design complexity (Lambright and Ume 1996). Moreover, significant functional and technical barriers often prevent the free flow of the necessary knowledge and information (Forgionne 1994). Mathematical programming, utility analysis and algorithm-rigorous optimization modeling approaches (*e. g.* compromise decision support problem (cDSP) and goal programming techniques) are data and information based, and thus cannot handle knowledge by nature. They are only for quantitative (tangible) criteria but not for qualitative (intangible) criteria (difficult to quantify). A knowledge-based decision support model, however, as proposed here, overcomes many of the shortcomings discussed earlier.

16.3 Design Decision Support Process

16.3.1 Decision Support Process

Generally speaking, decision is a choice, which is to realize a certain goal by analyzing subjective-objective conditions, generating alternatives, and choosing the most appropriate one among them. A generic decision support process can be described as having the following interactive aspects: intelligence, design, choice and implementation, as shown in Figure 16.1.

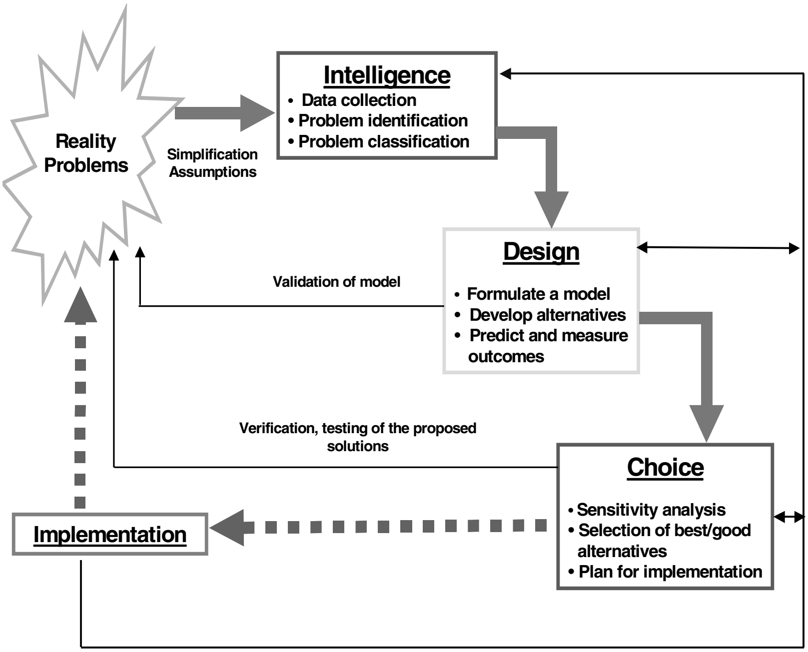


Figure 16.1. Decision support process (from Simon 1976)

It experiences the stages ranging from problem identification and classification, simplification of assumptions, data collection, model formulation, solution alternatives generation, evaluation, and selection, as well as model validation and verification and testing of the proposed solution to final plan implementation. The current research is focused on how knowledge support can aid the decision maker to make a decision during the design process. Figure 16.2 illustrates a scenario of implementing knowledge-based decision support (DDS) from the perspective of decision knowledge management (DKM), in which knowledge management technologies include knowledge generation and acquisition, knowledge codification, and knowledge processing and utilization (reasoning), *etc.*

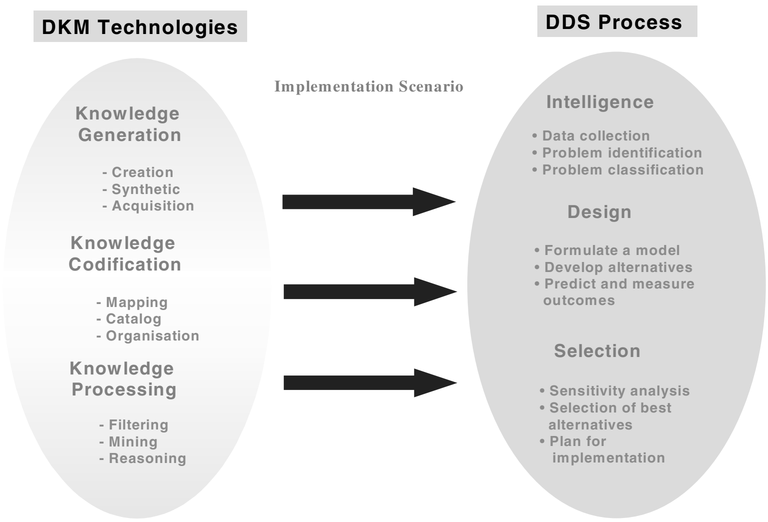


Figure 16.2. Decision support implementation scenario

16.3.2 Decision-based Design Process

The main role of a designer is to apply scientific and engineering knowledge to find (generate, evaluate and select) the solutions of design problems, and then optimize those solutions within the framework composed requirements and constraints set by physical, environmental and human-related considerations. We view design as the process of converting information that characterizes the needs and requirements for a product into knowledge about a product. Based on the principle of decision-based design, design equation can be expressed as follows (Mistree *et al.* 1995): $\{\boldsymbol{K}\}=\boldsymbol{T}\{\boldsymbol{I}\}$, where, $\boldsymbol{K}$ is knowledge output, $\boldsymbol{I}$ is information input, and $\boldsymbol{T}$ is transformation relationship, respectively. Thus, knowledge-intensive support becomes more critical in the design process and has been recognized as a key enabling technology for retaining a competitive advantage in product development.

In this chapter, we present the development of a knowledge-intensive design decision support scheme, as depicted in Figure 16.3, in which design decision support is exploited from the synthesis of design process modeling (DPM), knowledge management (KM), and decision support (DS). From the motivations and an overview of the design decision- making support process, it can be seen that the decision theories for example game theory, utility theory, probability theory, fuzzy set theory and extension set theory, *etc.* play a key role during the process (see Hazelrigg 1996 for discussion of some of these techniques).

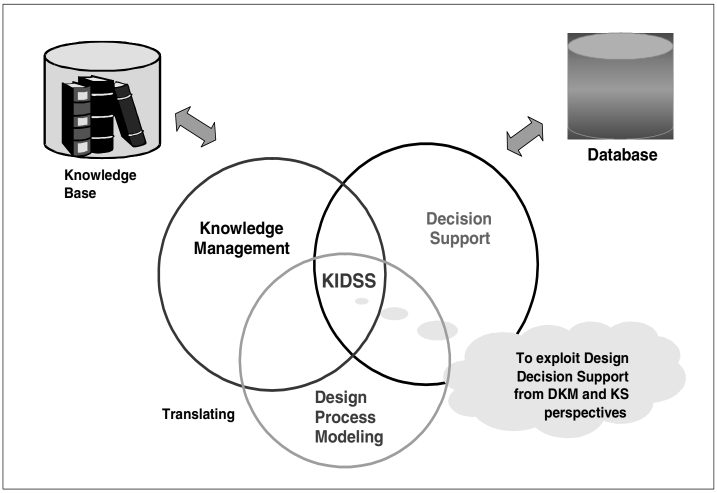


Figure 16.3. Knowledge-intensive design support system (KIDSS)

16.4 Hybrid Robust Decision Model

In this section, we establish a hybrid robust decision model that may integrate one or more techniques such as cDSP, fuzzy system, neural networks, intelligent agents, data mining and knowledge discovery (*e. g.* fuzzy clustering algorithm), extension theory and genetic algorithm, *etc.*, to solve both cooperative and noncooperative, compatible and incompatible decision problems. Details of these techniques are provided below.

16.4.1 Compromise Decision Support Model (cDSP)

Decision support problems (DSPs) are generally formulated using a combination of analysis-based hard information and engineering judgment in the form of viewpoints, postsolution sensitivity analysis, bounds, and context for decisions to be made. Two primary types of decisions are supported within the DSP technique: selection and compromise, and along with several combinations of these. The "selection" type decision actually includes evaluation and indication of preference based on multiple attributes for one among several feasible alternatives, while the "compromise" type decision is the improvement of a given alternative through modification. Another aspect of the DSP technique that is particularly relevant to distributed collaborative design is the facility of expressing decisions that are linked together such as coupled and hierarchical decisions through combinations of selection and compromise DSPs (*i. e.* selection-selection, compromise- compromise, and selection-compromise) (Xiao *et al.* 2002). These derived decision constructs are ideally suited for modeling networks of concurrent and sequential decisions that share information and knowledge. In the compromise decision support problem (cDSP) model, a hybrid of goal programming and mathematical programming is used to determine the values of design variables that satisfy a set of constraints and achieve as closely as possible a set of conflicting goals. For more details, please refer to (Mistree *et al.* 1993, 1995).

16.4.2 Fuzzy Synthetic Decision Model (FSD)

The problem of design evaluation and selection can be defined as: given a set of design alternatives, evaluate and select a design alternative that can satisfy customer needs, meet design requirements and fit the technical capabilities of a company. To combine expert judgment and process useful knowledge for decision-making, a fuzzy synthetic decision model is developed based on fuzzy AHP, ranking algorithms and inference mechanisms for engineering design evaluation and selection.

16.4.2.1 Fuzzy Analytic Hierarchy Process

The AHP mechanism proposed by Saaty (1991) is widely recognized as a useful tool to support multiattribute decision making. It is a compositional approach where a multiattribute problem is first structured into a hierarchy of interrelated elements, and then a pairwise comparison of elements in terms of their dominance is elicited. The weights are given by the eigenvector associated with the highest eigenvalue of the reciprocal ration matrix of pairwise comparisons. Using AHP, a designer is capable of choosing weights by comparing the importance of two criteria subjectively. The pairwise comparison ratio which is the comparison of the importance of criterion $\boldsymbol{i}$ and criterion $\boldsymbol{j}$, that is $\boldsymbol{w_i}$ i and $\boldsymbol{w_j}$, is defined as:

$$\boldsymbol{a_{ij}} = \boldsymbol{w_i} / \boldsymbol{w_j} \tag{16.1}$$

Considering a pairwise comparison matrix A = [$\boldsymbol{a_{ij}}$] and an importance index (weight) vector $\boldsymbol{W}$ = [$\boldsymbol{w_i}$], their relationship can be described according to:

$$\boldsymbol{AW} = \boldsymbol{nW} \tag{16.2}$$

When $\boldsymbol{A}$ is given, $\boldsymbol{W}$ and $\boldsymbol{n}$ are calculated as an eigenvector and an eigenvalue of $\boldsymbol{A}$, respectively. In this study, each agent has its own matrix $\boldsymbol{A}$, and exchanges the matrix between agents to cooperatively adapt to changes in the design process. In AHP, the pairwise comparison matrix should be examined for reliability of consistency. The consistency index ($\boldsymbol{CI}$) is calculated as:

$$\boldsymbol{CI} = \frac{\lambda_{\max} - n}{n - 1} \tag{16.3}$$

where λ_{max}, is the maximum value of 0. If the value of $\boldsymbol{CI}$ is higher than 0.1, the matrix should be reset by comparing importance again. Therefore, we should focus on the comparison matrix $\boldsymbol{A}$. Currently, most of researchers compose AHP comparison matrix $\boldsymbol{A}$ according to user's individual and flexible preferences. In a flexible negotiation environment, however, most of agents may change their offers according to counter offers. Hence, there is a need to build the comparison matrix ***A*** dynamically. In this work, we combine fuzzy membership functions with the AHP to pursue the preference of agents dynamically, and as a result, we propose the fuzzy comparison matrix ***A***.

16.4.2.2 Fuzzy Ranking for Evaluation

Using the design solution clustering techniques (*e.g.* cDSP model above) at the conceptual design stage, a reasonable number of possible design alternatives can be obtained. Once this is achieved, one needs to examine the design alternatives against marketing and econotechnical as well as ergonomic criteria and aesthetic criteria. This is actually a multicriteria decision-making problem. One of the well-known methods for multicriteria decision making is the traditional procedure for calculating a weighted average rating $\overline{r_i}$ by use of value analysis or cost-benefit analysis (Pahl and Beitz 1996):

$$\overline{r}_i = \sum_{j=1}^{n}(w_j r_{ij}) / \sum_{j=1}^{n} w_j \tag{16.4}$$

where, i=1,2,…,m, j=1,2,3,…, n , r_{ij} denotes the merit of alternative a_i according to the criterion C_j; w_j denotes the importance of criterion C_j in the evaluation of alternatives. The higher $\overline{r_i}$, the better its aggregated performance.

However, the above traditional procedure is not applicable for situations where uncertainty exists and the available information is incomplete. For example, the terms “very important,” “good,” or “not good” themselves constitute a fuzzy set. Here, we give an example of the problem of fuzzy ranking in terms of evaluating a set of alternatives against a set of criteria (Zadeh 1965, Kickert 1978, Gui 1993). Let a set of m alternatives $A=\{a_1, a_2,...,a_m\}$ be a fuzzy set on a set of n criteria $C=\{C_1, C_2,..., C_n\}$ to be evaluated. Suppose that the fuzzy rating $\tilde{r}_{ij}$ to certain C_j of alternative a_i is characterized by a membership function $\mu_{\tilde{R}_{ij}}(\tilde{r}_{ij})$, where, $\tilde{r}_{ij} \in R$, and a set of weights $\tilde{W} = \{\tilde{w}_1, \tilde{w}_2,..., \tilde{w}_n\}$ is fuzzy linguistic variables characterized by $\mu_{\tilde{W}_j}(\tilde{w}_j), \tilde{w}_j \in R^+$. Consider the mapping function $g_i(\tilde{z}_i): R^{2n} \to R$ defined by:

$$g_i(\tilde{z}_i) = \sum_{j=1}^{n}(\tilde{w}_j \tilde{r}_{ij}) / \sum_{j=1}^{n} \tilde{w}_j \tag{16.5}$$

where, $\wedge^{\circ}$ is the calculation operator of taking the minimum. Thus, through the mapping $g_i(z_i): R^{2n} \to R$, the fuzzy set $\tilde{Z}_i$ induces a fuzzy rating set $\tilde{R}_i$ with the membership function

$$\mu_{\tilde{R}_i}(\tilde{r}_i) = \sup_{Z_i g(\tilde{z}_i)=\tilde{r}_i} \mu_{\tilde{Z}_i}(\tilde{z}_i), \tilde{r}_i \in R \tag{16.6}$$

The final fuzzy rating of design alternative a_i can be characterized by this membership function. But it does not mean the alternative with the maximal $\mu_{\tilde{R}}(\tilde{r}_i)$ is the best one. The following procedure can be employed to further characterize the two fuzzy sets as (Gui 1993):

(1) a conditional fuzzy set is defined with the membership function:

$$\mu_{I/R}(i \mid \tilde{r}_1,...\tilde{r}_m) = \begin{cases} 1 & \text{if} \quad \tilde{r}_i > \tilde{r}_k, \forall k \in (1,2,...,m) \\ 0 & \text{otherwise} \end{cases} \tag{16.7}$$

(2) a fuzzy set is constructed with membership function:

$$\mu_R(\tilde{r}_1,...\tilde{r}_m) = \overset{\circ}{\bigwedge}_{i=1,...,m} \mu_{\tilde{R}_i}(\tilde{r}_i) \tag{16.8}$$

A combination of these two fuzzy sets induces a fuzzy set in which one can determine a best design alternative with the highest final rating, *i. e.*,

$$\mu_I(i) = \sup_{\tilde{r}_1,...\tilde{r}_m} \mu_{I/R}(i \mid \tilde{r}_1,...\tilde{r}_m) \overset{\circ}{\wedge} \mu_R(\tilde{r}_1,...\tilde{r}_m) \tag{16.9}$$

Comparing with (16.4), the fuzzy ranking for design alternatives is more flexible and presents uncertainty better. Based on this method, a designer can now effectively and consistently incorporate linguistic rating and weights such as “good,” “fair,” “important,” “rather important,” *etc.*, in design alternatives evaluation.

16.4.2.3 Evaluation Function and Index for Selection

The design space for a complex system is very large. The designer is often required to consider not only the product functionality, but also other criteria including compactness and other life-cycle issues, such as manufacturability, maintainability, reliability, and efficiency. Some of these criteria may contradict each other. Designers should analyze the tradeoffs among various criteria and make the “best” selection from the available alternatives. As such, it is important to have a powerful search strategy that will lead to a near optimum solution in a reasonable amount of time. The A* search algorithm constitutes such a method (Sriram 1997). In the proposed approach, the system first calculates the weighted performance rating aggregation of each retrieved alternative by analyzing the tradeoff among various criteria. Then, it calculates the evaluation index of each design alternative by considering all the weighted performance ratings. After calculating the numerically weighted performance ratings of all design alternatives, the evaluation index is used as a heuristic evaluation function f_h, by considering all the weighted performance ratings $\bar{r}_i$ (i=1,2, …, m) of its constituent members and the number $\boldsymbol{k}$ of its unsatisfied customer requirements, as follows:

$$f_h = \sum_{i=1}^{m}(1/\bar{r}_i) + k \qquad (16.10)$$

where, $\bar{r}_i \in [0,1]$ is the numerical weighted performance rating of the design alternative $\boldsymbol{a}_\mathbf{i}$; $1/\bar{r}_i = (1,+\infty)$ is defined as the performance cost of design alternative $\boldsymbol{a}_\mathbf{i}$. A higher-weighted performance rating of a design alternative corresponds to a lower performance cost. $\sum_{i=1}^{m}(1/\bar{r}_i)$ represents the accumulated performance cost of a design alternative along the search path thus far. $\boldsymbol{k}$ is a heuristic estimate of the minimal remaining performance cost of a design alternative along all the possible succeeding search paths. f_h is the estimate of the total performance costs of a design alternative. f_h is also called the evaluation index or the heuristic evaluation function. In (16.10), a higher $\bar{r}_i$, *i. e.* a better-aggregated performance of each retrieved design alternative $\boldsymbol{a}_\mathbf{i}$, and a lower $\boldsymbol{m}$ or $\boldsymbol{k}$, *i. e.* a higher compactness of a design alternative, will result in a lower evaluation index of a design alternative f_h. Thus, at each step of the A* search process, the best design alternative, *i. e.* the one with the lowest value of the heuristic evaluation function is selected, by taking into account multi-criteria factors including design compactness and other life-cycle issues, such as manufacturability, assemblability, maintainability, reliability, and efficiency.

16.4.3 Integration and Cooperation of Decision Models

All available algorithms for optimization and constraint satisfaction have weaknesses; more rigorous algorithms tend to be too slow, heuristics, too unreliable. Rather than attempting to design a new algorithm without weaknesses, a task that is difficult if not impossible, some researchers have been working on ways to organize algorithms so that they can suppress their respective weaknesses through cooperation, and together achieve what separately they might not (Talukdar *et al.* 1996, Zha 2003). As stated above, the cDSP model is basically data and information centric and more appropriate for implementation in conjunction with tangible (quantitative) criteria rather than for intangible (qualitative) criteria. The FSD model is knowledge based and able to handle both intangible and tangible criteria (*e. g.*, from fuzzy requirements to crisp design). The synthesis of the cDSP and FSD models can generate a more powerful robust decision model. The scheme or mode of integration and coordination could be either “loose,” or “tight.” In the loose mode, two or more models are combined and they work together but complement each other. Depending on the nature of the decision problem, an adaptor is employed in the model and serves as a regulatory switch to adapt the decision problems by shifting the paradigms from one decision method (*e. g.* cDSP) to another (*e. g.* FSD). Together with a genetic algorithm (sGA), a systematic knowledge-based adjustment method for parameters is developed for the decision maker in the complex system design. The regulatory switch is implemented using sGA and the knowledge-based guidance (Lu *et al.* 2000). In the tight mode, two or more models coexist and are integrated into a single hybrid model, for example, fuzzy cDSP, fuzzy neural networks or the neurofuzzy system above, *etc.* Figure 16.4 provides a schematic view of the hybrid robust decision model integrating cDSP and FSD models.

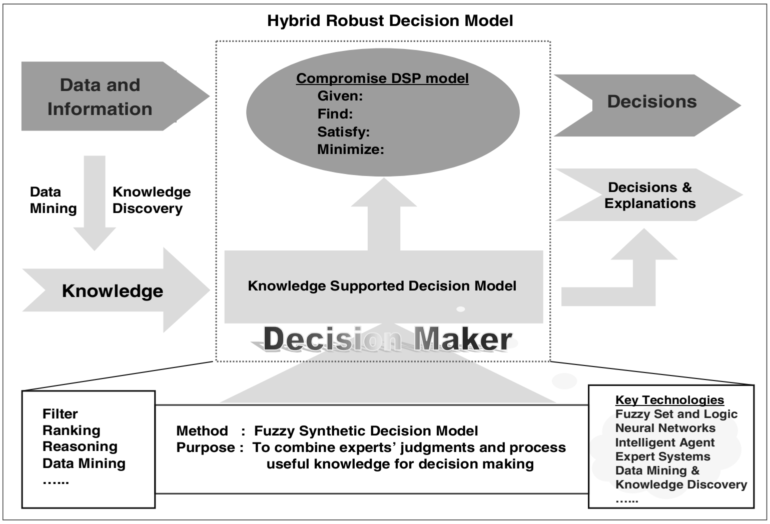


Figure 16.4. Hybrid robust decision support model

This kind of knowledge-based model can manage design decision knowledge and provide real-time or online support to designers in the decision-making process: 1) overcome shortcomings of cDSP; 2) suggest solutions and provide explanations to the designer; 3) may be used in the early design stage; and 4) stimulate the designer to generate new design ideas (with learning).

16.5 Collaborative Decision-making Mechanisms

Coordination is the central problem of multiagent systems (MAS), which includes cooperation and the conflict resolution problem. In fact, the conflict problem is noncooperative problem, so the conflict resolution is the key technology for MAS. The common ways to solve the conflicts are arbitration and negotiation. The arbitration is based on the classic mathematic theory and reasoning rules according to the concept of characteristic function which means that "to be or not to be, (yes, no) or (0,1)," while the negotiation means is based on fuzzy set theory and reasoning rules according to the concept of membership function which means that the membership's degree of "to be or not to be, [yes, no] or [0,1]." Both ways result in the cost of sacrificing individual agents' interests with different degrees. In this section, two mechanisms, the transforming bridge and regulatory switch (Zha *et al.* 2003), are used for solving conflict or incompatible problems and collaboration/negotiation between designers/ decision makers in design.

16.5.1 Transforming Bridge

The transforming-bridge method is proposed to deal with conflict or incompatible decision problems. Two disputing factions require a bridge to span the river that separates them. The river is the neutral territory and therefore the bridge must physically transform to serve as a place for conflict resolution. When transformed, the "room" must no longer function as a bridge. The typical imaginary example for using this method is how people in Hong Kong solve the so-called "connections between right side drive and left-side drive" problem. Thus, the concept of a transforming bridge enables the two incompatible sides to prevail in maintaining their own two specific interests. It is neither a solution method based on competition nor the one which tradeoff and balance are not their interests.

We deal with the conflict problems as opposite (compatible and incompatible) problems. Transformation is based on the reasoning rules according to the concept of dependent (transfer) function, which means "to be or not to be can be interchanged". Based on the principles of "go the contrary way" and "disport frequency of multiple branches" a transforming bridge can be devised to solve multiagent resource conflict problems and dynamic design conflict problems. Specifically, a design-transforming bridge, which can play a connected and transformable role, is designed between two players so that the conflict can be resolved and therefore two players can gain their satisfied solutions. By handling the incompatible problems, the searching range will be expanded step-by-step according to the degree of conflict of the to-be-solved problems. For example, the chess-playing process could be analyzed using this bridge. When more than one piece of chess is in danger, by normal optimal algorithms the smallest loss will be calculated and the piece with the smallest loss will be abandoned. By means of an extension strategy, the already calculated smallest loss will be used to extend the set and search for a new chance among those pieces that are momentarily not in danger but have an opportunity to start a new attack. The price of this new attack must be even smaller than the calculated loss before.

16.5.2 Regulatory Switch

Traditionally, the designer usually depends on a human's knowledge and trial-and-error when determining a parameter value. However, these methods are not easy to apply when there are too many system parameters with potential relationships. A genetic algorithm has the advantage of searching optimum and avoiding local values. Together with a genetic algorithm, a systematic adjustment method for parameters is developed for a decision maker in a complex system design. The sGA

(Dasgupta and McGregor 1994) design representation uses regulatory genes that act as a switch to turn genes on (active) and off (passive). Each gene in higher levels acts as a switchable pointer that has two possible targets: when the gene is active (on) it points to its lower-level target (gene), and when passive (off) it points to the same-level target. At the evaluation stage only the expressed genes of an individual are translated into the phenotypic functionality, which means that only the genes that are currently active contribute to the fitness of the decision. The passive genes do not influence fitness and are carried along as redundant genetic material during the evolutionary decision-making process. Therefore, the utilization of the sGA approach to collaborative design decision can be summarized as follows. First, genes represent decision modules or subsystems that are either active or passive, depending on whether or not they contribute to the decision problem. Then, a family of decision solutions relied on the addition or subtraction of decision modules could be evaluated by alternating different "active" and "passive" modules or subsystems. A family of solutions would thus correspond to decision model variants that have different active and passive combinations of modules or subsystems.

16.5.3 Negotiation Support

During the decision making between multiple designers, it is crucial to negotiate on multiple attributes of a design deal such as material, manufacturing method, parameters values, cost, quantity, quality, and relative preference. The negotiation is a form of decision making with two or more actively involved agents who could not make decisions independently, and therefore must make concessions to achieve a compromise. Therefore, negotiations for an enormous volume of transaction on the Internet became a fundamental mechanism to automating collaborative design. Furthermore, the flexibility and adaptability of the negotiation mechanism may be used as a plausible source of motivation and framework for the design of intelligent and autonomous agent systems (Kim *et al.* 2003). In this work, the negotiation mechanism using the FSD model is composed of the following six phases:

(1) The negotiation mechanism is started with the 'initial offers for a design deal' of agent (designer). In this phase, each negotiation agent offers their negotiation conditions reflecting their relative preference for a deal. The design deal is composed of quantitative conditions such as the parameters values and cost. However, the fuzzy values for these conditions are changed by fuzzy membership functions reflecting qualitative conditions such as relative preference.

(2) After the initial offers of agent, 'fuzzy membership functions' are used to support the construction of fuzzy pairwise comparison matrix A. Using this fuzzy membership function, designer's relative preferences are transformed into fuzzy membership values. During the transformation process, bell-shaped (or Z, λ, π and S-type) fuzzy membership functions can be adopted.

(3) The ‘pairwise comparison matrix A’ is constructed. In this phase, the AHP comparison matrix is used to compute the relative importance about each alternative (deal). As a result, each agent’s offers are fully compared.

(4) This phase is for ‘selection of preferred offer’ of the negotiation agent. Based on the result of comparison in phase (3), the preferred offers are selected by one or some designers. However, this is the first step of the dynamic negotiation process.

(5) The fifth phase is for ‘revision of offers and negotiation’. In this phase, each agent revises their ‘initial offer’ and continues to negotiate with their counterpart. For this purpose, the ‘goal-seeking’ methodology is used to revise the initial offers.

(6) The final phase is to suggest the ‘optimal offer’. The fuzzified pairwise comparison matrix A and the AHP inference mechanism are used to suggest the optimal offer, and then go to the phase (5) to lead a consensus with their counterpart. As a result, each designer could be satisfied with the final offers.

16.6 Multiagent Collaborative Design Decision Support Framework

The overall knowledge-intensive multiagent design decision support scheme proposed in (Zha 2003) is shown in Figure16.5. This scheme consists of a design process modeling and management agent, a knowledge capture agent, a knowledge repository, codesigners, a decision support agent, *etc.* The communication, negotiation and execution mechanisms between these agents are modeled with contract nets. The core of the scheme is the decision support agent that is the focus of discussion. The knowledge repository is used to store, share, and reuse the corporate design knowledge (Szykman *et al.* 2001). A prototype web-based design decision support system has been developed to verify and demonstrate the developed methodologies (algorithms) and framework. The decision support agent could be used as an autonomous agent to be finally integrated into a web-based product design and realization framework to support collaborative decision-making in the product development process (design chain). The decision support agent should be able to make autonomous decisions concerning: 1) spawning an agent to search in a given direction, 2) killing an agent that is not very successful, 3) negotiation between agents (unless they need to consult the designer), 4) recognition of novelty of a solution (eventually consulting the knowledge repository or database of existing solutions) and turning the designer’s attention towards it, 5) when to consult the designer, *etc.*

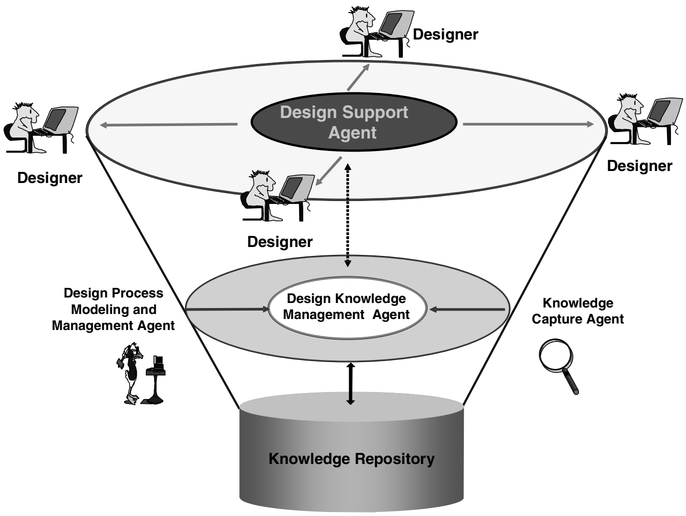


Figure 16.5. The overall multiagent knowledge intensive decision support framework

The comparative ranking of alternatives and decision making discussed in Section 16.4.2 is a fundamental component of the design decision agent. As stated previously several formal decision models exist (Section 16.2). Utility theory (Keeney and Raiffa 1976) and AHP (Saaty 1991) are well-known examples. The decision agent, illustrated in Figure 16.6, is a container specialized in providing evaluation services. It contains criteria that pair design attributes (variable modules) with preference modules (a type of variable module used to define preference functions). The decision agent provides an overall multiple attribute evaluation service while each criterion evaluates a single-attribute. The relations of the criterion and decision agent are not user defined. The criterion relations calculate the worth of the design attribute based upon the preference model, while decision agents automatically generate relations to aggregate single attribute evaluations for multiple attribute decision. Thus, there are different types of decision agents for each decision theory. In the prototype implementation the decision agent has been developed by integrating the cDSP technique with an expert/knowledge model into a hybrid robust decision support model for criterion/argument analysis and fusion.

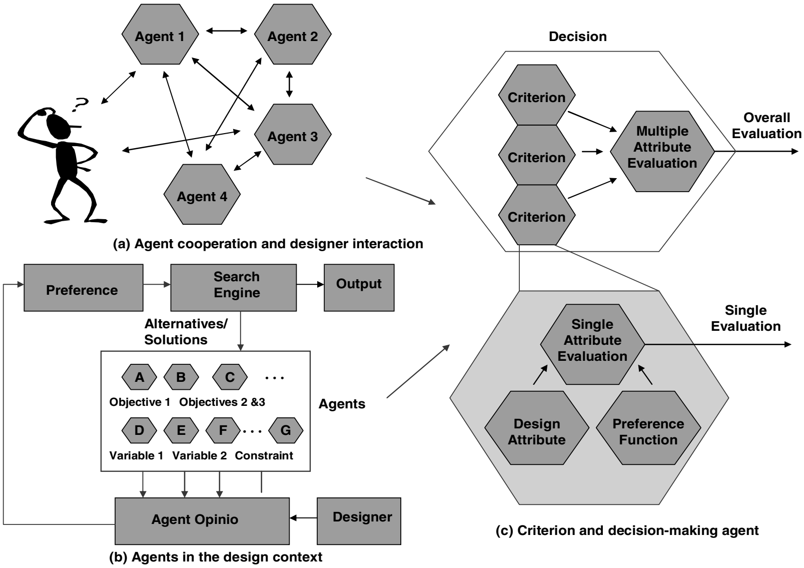


Figure 16.6. Decision support agent

16.7 Application in Conceptual Design Decision Support

During the conceptual design stage (Figure 16.7), a family of product concepts (or product concept variants) can vary widely by the selection and assembly of modules or predefined building blocks at different levels of abstraction so as to satisfy diverse customers' requirements. A wrong or even a poor selection of either a building block or module can rarely be compensated for at later design stages and can give rise to a great expense of redesign costs (Pahl and Beitz, 1996). Thus, concept evaluation and selection is crucial in this stage. We propose a knowledge decision support approach to concept evaluation and selection, as shown in Figure 16.8. The kernel of the knowledge decision support scheme is the hybrid decision support model discussed above. This model is used for design concept evaluation and selection, in which the cDSP model is used to cluster/classify design alternatives or variants and determine similarity and commonality between modules, product variants and product families; while the FSD model is used to evaluate and select a design alternative that satisfies customer needs, meets design requirements and complies with the technical capabilities of a company. The knowledge resource utilized in the process extensively includes differentiating features, customers' requirements, desires, preferences and importance (weights), tradeoffs (*e. g.* market *vs*. investment), and utility functions, and heuristic knowledge, rules, *etc.*

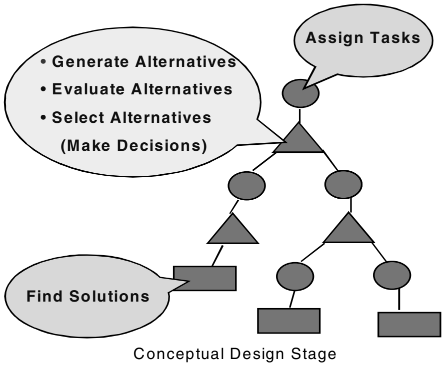


Figure 16.7. Concept evaluation and selection in design

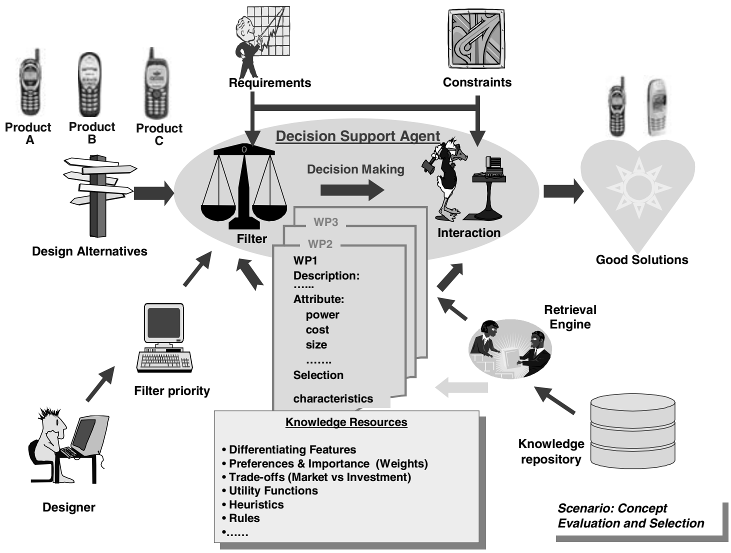
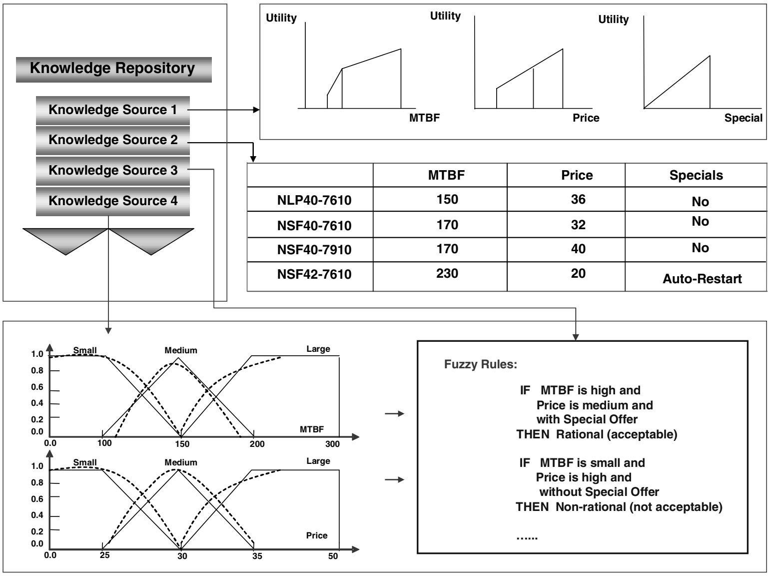


Figure 16.8. Knowledge decision support for concept evaluation and selection

16.8 Case Study

The above proposed knowledge-support scheme has been used for decision making in a power supply family design for customization. Specifically, the cDSP model is used to cluster/classify power-supply family product design alternatives or variants and determine similarity and commonality between modules and product variants. The FSD model is used to evaluate and select the power-supply design alternatives that satisfy customer needs, meet design requirements and comply with the technical capabilities, in which the negotiation is involved.

From a customer's point of view, a power supply product is defined based on the following required features (RFs): power, output voltage (OutV), output current (OutC), size, regulation, mean time between failure (MTBF), *etc.* From an engineers' point of view, the power supply product is designed by determining these variables (parameters) (DPs): core of transformer (core), coil of transformer (coil), switch frequency (switchF), rectifier, heat sink type (typeHS), heat sink size (sizeHS), control loop (control), *etc.* Using the cDSP model and fuzzy clustering, three different clusters are obtained. Three product families I, II and III are generated based on three different clusters, which have 4, 5 and 3 base products (BPs) respectively. Each cluster has its own range/limitation with regard to particular product features and/or design parameters. When the product configuration is carried out, the design requirements and constraints are satisfied especially in terms of product functions or functional features.



	MTBF	Price	Specials
NLP40-7610	150	36	No
NSF40-7610	170	32	No
NSF40-7910	170	40	No
NSF42-7610	230	20	Auto-Restart

Figure 16.9. Knowledge used in power-supply product design for customization decision

With reference to the knowledge decision support scheme for product evaluation, a scenario of knowledge support for power-supply product evaluation for customization in Family I is shown in. The customers' requirements for Family-I power supplies include AC/DC, 45 W, 5 V and ±15 V, 150 kh, \$20-50, *etc.* The knowledge decision support system first eliminates unacceptable alternatives and determines four acceptable alternatives: NLP40-7610, NFS40-7610, NFS40-7910, and NFS 42-7610. The final design decision can be reached based on the knowledge resources given in Figure 16.9, including differentiating features (MTBF, price, and special offer) and their utility/membership functions, fuzzy rules, *etc.* The final design decision made by the system is NFS42-7610 as it has maximum MTBF, medium price and special offer of auto-start function, and it is acceptable based on the rules.

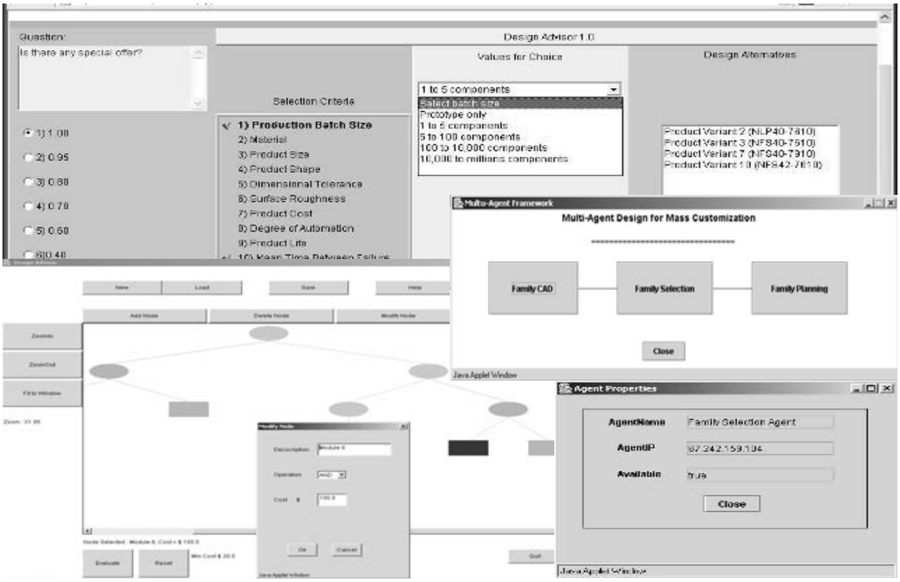


Figure 16.10. Screen snapshot for power-supply product evaluation and selection for customization

Table 16.1. Weights and partial performance ratings

Criterion No.	Criterion Item	Criterion Weight			Partial Performance Rating		
		Linguistic Term	Fuzzy Number	Weight Value	Linguistic Term	Fuzzy Number	Rating Crisp Value
1	MTBF	High	(0.7,0.8,0.8,0.9)	w_1=0.80	Medium	(0.4,0.5,0.5,0.6)	r_{11}=0.500
2	Price	Fairly High	(0.5,0.6,0.7,0.8)	w_2=0.65	High	(0.7,0.8,0.8,0.9)	r_{12}=0.800
3	Special Offer	Medium	(0.4,0.5,0.5,0.6)	w_3=0.50	Very Low	(0.0, 0.0,0.1,0.2)	r_{13}=0.075

Evaluation Results:

No.	Family I	Evaluation Index (h)	Rankings
1	NLP40-7610	2.128	3
2	NFS40-7610	2.041	2
3	NFS40-7910	2.222	4
4	NFS42-7610	1.449	1

Table 16.1 gives weights and partial performance ratings for each criterion (for NLP40-7610) and evaluation results. Figure 16.10 shows a screen snapshot for the power-supply product evaluation and selection for customization.

16.9 Conclusions and Future Work

In this chapter we presented a hybrid decision model and a multiagent framework for collaborative decision support in the design process. The hybrid decision model presented in this chapter provides a clean and effective digital interface and design decision templates for a series of decisions in design process in the knowledge intensive and distributed collaborative environment. The knowledge-based decision support model can manage design decision knowledge and provide real-time or on-line knowledge support to designers in the decision-making process. It can compensate for typical barriers to the decision-making process, including incomplete and evolving information, uncertain evaluations, inconsistency of team members' inputs, *etc*. The robust decision assessment process can be used and refined for the product development process mapping, constraint and gap identification, tracking the information development and flow, and measuring the effectiveness of current processes. Designers, especially novices, can benefit from retrieval of knowledge about previous designs by abstracting information and applying it to a new design or by gaining insight into how an earlier related product was designed. By making use of the design knowledge, companies are expected to improve the design process for more innovative products and reducing product development cycle time. As a kernel of the knowledge-supported design system, the design decision support system (agent) can help design teams make better decisions. The application in concept evaluation and selection in design for mass customization illustrates the feasibility and potentials of the developed methodology and framework. The developed methodology is flexible enough to be used in a variety of decision problems.

Disclaimer

Commercial equipment and software, many of which are either registered or trademarked, are identified in order to adequately specify certain procedures. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

References

Dasgupta D, McGregor DR (1994) A more biologically motivated genetic algorithm: the model and some results. Cybernetics and Systems: An International Journal 25: 447-469

Forgionne GA (1994) Decision technology system to deliver effective concurrent engineering. Concurrent Engineering: Research and Applications 2(2): 67-76.

Gui JK (1993) Methodology for Modeling Complete Product Assemblies, PhD Dissertation, Helsinki University of Technology, Finland

Hazelrigg G (1996) System Engineering: An Approach to Information-based Design, Prentice-Hall International Series in Industrial and System Engineering

Jiao JX, Tseng MM (1998) Fuzzy ranking for concept evaluation in configuration design for mass customization. Concurrent Engineering: Research and Application 6(3):189-206

Keeney RL, Raiffa H (1993) Decisions with Multiple Objectives: Preferences and Value Tradeoffs. Cambridge University Press

Kickert WJM (1978) Fuzzy Theories on Decision-making: A Critical Review, Martinus Nijhoff Social Sciences Division

Kim JS (2003) Negotiation support in electronic commerce using fuzzy membership functions and AHP, Proceedings of the 6th Pacific Rim International Workshop on Multi-Agents (PRIMA) 2003, Seoul (Korea), pp.93-104

Lu J, Quaddus MA, Williams R (2000) Developing a Knowledge–based Multi-Objective Decision Support System, Proceedings of the 33rd Hawaii International Conference on System Sciences, pp. 1-10

Lambright JP, Ume C (1996) A flat composite panel design advisory system using knowledge based and case based reasoning. Journal of Mechanical Design 118 December: 461-475.

Mistree F, Hughes OF, Bras BA (1993) The Compromise Decision Support Problem and the Adaptive Linear Programming Algorithm, Structural Optimization: Status and Promise, M.P. Kamatt (ed.), AIAA, Washington D. C., Chapter 11, pp 247-286

Mistree F, Bras B, Smith WF, Allen JK (1995) Modeling design processes: a conceptual decision-based perspective. Engineering Design & Automation, 1(4): 209-321

Pahl G, Beitz W (1996) Engineering Design - A Systematic Approach, New York: Springer

Rosen DW, Chen Y, Gerhard J, Allen JK, Mistree F (2000) Design Decision Templates and Their Implementation for Distributed Design and Solid Freeform Fabrication, Proceedings of ASME DETC00, Chapter No. DETC00/DAC-14293, Baltimore, Maryland, September 10-13

Sriram RD (1997) Intelligent Systems for Engineering: A Knowledge-based Approach. London: Springer Verlag, UK

Sriram RD (2002) Distributed and Integrated Collaborative Engineering Design. Sarven Publishers, Glenwood, MD, USA

Suh NP (1990) The Principles of Design. Oxford University Press, New York, NY

Saaty TL (1991) The Analytic Hierarchy Process. McGraw-Hill, New York, NY

Simon HA (1976) Administrative Behavior. Free Press, New York

Szykman S, Sriram RD, Regli W (2001) The role of knowledge in next-generation product development system. ASME Journal of Computing and Information Science in Engineering 1(1): 3-11

Talukdar S, Baerentzen L, Gove A, de Souza P (1996) Asynchronous Teams: Cooperation Schemes for Autonomous Agents. Carnegie Mellon University, Pittsburgh, PA 15213

Xiao A, Choi H, Allen JK, Rosen DW, Mistree F (2002) Collaborative Decision-making Across Digital Interfaces, Proceedings of ASME DETC 2002, Paper No.: DETC2002/ DAC-34073, Montreal, Canada

Zadeh LA (1965) Fuzzy sets. Information and Control 8: 338-353

Zha XF, (2003) Knowledge intensive decision support for design process: a hybrid robust model and framework. Proceedings of ICED 03, Stockholm, Sweden

Zha XF, Sriram RD, Lu WF (2003) Knowledge Intensive Collaborative Decision Support for Design Process. Proceedings of ASME DETC 2003, Paper No: DETC2003/DAC-48747, Chicago

17

The Application of Semantic Web Technologies for Railway Decision Support

Jinwei Lu[1], Clive Roberts[1], Karl Lang[2], Alan Stirling[1] and Keith Madelin[1]

[1] School of Engineering, The University of Birmingham, Birmingham, B15 2TT, UK, e-mail: j.lu@bham.ac.uk
[2] Department of Computer Information Systems, Baruch College, City University of New York, New York, 10010, USA, e-mail: karl_lang@baruch.cuny.edu

In this chapter, the potential of using Semantic Web technologies as part of an intelligent decision support system (DSS) is discussed. The Semantic Web is introduced and its benefits to DSSs are highlighted. In addition, two Semantic Web ontologies for DSSs are developed using the Protégé tool. The two ontologies are used to demonstrate their ability to infer new knowledge, help visualize and improve data presentation, query data and allow global database linkage as a part of the Semantic Web; this allows the ontologies to be processed by intelligent, Web-based software agents.

17.1 What is the Semantic Web?

The ability to make decisions is a crucial process in steering the course of national and international economics and defence strategy, as well as, business activities and personal lives. Decision making involves searching for conditions that call for decisions, and then designing and choosing actions. The quality of decision-making depends on both the decision makers and any decision support system (DSS) used (Simon *et al.* 1986, Turban 1995). DSSs are interactive computer-based systems that aid decision makers to gain a greater understanding of their businesses by taking advantages of both human intelligence and the ability of computers to access large quantities of data, develop models, interpret results and formulate and evaluate alternative solutions (Turban 1995).

In the UK, the privatization of the railway network has resulted in the separation of responsibility for various technical issues of the network, especially at the interface between the vehicle and track. Thus, it is felt that an intelligent, Internet-based railway DSS for improving the management and performance of the vehicle and track system is required.

The railway industry needs an internet based DSS to provide both a means of integrating analysis tools relating to vehicle and track and to provide access to railway expertise. For example, in the railway domain, the vehicle designer needs to understand the effects that changes to the vehicle speed, design parameters of the wheel and the rail, manufacturing processes, *etc.* have on vehicle derailment, ride performance and wear (Stribersky *et al.* 2004). The railway network operator needs to balance the network performance against maintenance costs. The railway systems engineer needs to know the impact one component of the system has on the others. Over two hundred vehicle and track analysis tools have been identified (Bayati *et al.* 2002) and the demand for a DSS to select an appropriate tool, or suite of tools, in order to help solve a given problem is obvious. In addition, a vast quantity of railway expertise and network standards are available over the Internet, but are currently not collated or linked. With the rapid development of internet technology there has been an increase in the use of DSSs in many organizations (Shim *et al.* 2002).

A large comprehensive knowledge base (KB), as a component of the previously mentioned DSS needs to be created for rail vehicle and track system integration. The KB is intended to query standards, patents and experts and serve the railway designer, manufacturer, maintainer, operator, *etc.*

The Semantic Web makes it possible to query or infer knowledge on a global scale using the Internet. Semantic Web ontologies are used to develop a framework for the creation of KBs. The Semantic Web is an extension of the current Web through the use of well-defined, linking information so that software agents can identify, interpret, manipulate, and interoperate the marked information among themselves (Hendler *et al.* 2002). The Semantic Web was initially specified as a global database, however, with the arrival of the OWL (Web Ontology Language) (W3C, 2004) additional functionality is possible. An ontology is a model of a specific domain of knowledge. A Semantic Web ontology is represented as classes together with their properties and instances. These can be managed in software packages such as Protégé, which have been developed specifically for Semantic Web ontology development (Protégé 2005). With the advent of standards and development tools such as OWL and Protégé, numerous applications of Semantic Web ontologies have been implemented in the areas of intelligent reasoning, information discovery, decision support, data fusion, systems integration and evolution of human knowledge (Rubin *et al.* 2004, Currie and Parmadee 2004, Flynn and Dean 2002a, Kogut and Heflin 2003, Berners-Lee *et al.* 2001). The Semantic Web ontology has become a core in developing frame-based KBs and subsequent applications.

This chapter will introduce Semantic Web technologies in the context of intelligent decision support and will create two domain specific ontologies that are able to provide intelligent reasoning. Section 17.2 provides an overview of the key concepts of the Semantic Web through the explanation of the DAML (DARPA Agent Markup Language) project for creating and measuring Semantic Web technologies, the Wine Agent that acts as a test-bed application for the development of the Semantic Web and the OWL and Protégé software. In addition, the benefits that Semantic Web technologies can bring to the community of decision support are highlighted. Section 17.3 provides two railway-domain ontologies. Finally a summary and conclusions are provided in Section 17.4.

17.2 The Semantic Web and its Benefits to DSS

17.2.1 Introduction to the Semantic Web

Berners-Lee *et al.* (2001) developed the idea of the Semantic Web to aid with Web semantic interoperability. They developed a personal, multicriteria decision-making scenario where clinic appointments were to be set up using software agents to access structured information and inference rules were used over the Internet, which in turn reasoned and presented alternatives. From this scenario, many essential requirements of the Semantic Web were conceived, such as a generic machine interpretable definition of Web content, logic representations of knowledge so as to make inference, the ability to choose courses of action and answer questions and the specification for agents to collect and process information and interoperate among themselves.

> *The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation. It is based on the idea of having data on the Web defined and linked such that it can be used for more effective discovery, automation, integration, and reuse across various applications. For the Web to reach its full potential, it must evolve into this Semantic Web, providing a universally accessible platform that allows data to be shared and processed by automated tools as well as by people* (Hendler *et al.* 2002).

Kogut and Heflin extended the Semantic Web to legacy systems with similar motivations and mechanisms. Agent-based systems were regarded as a powerful treatment for intensive information overload and as a solution to interoperability problems between legacy systems and other Web services. However, agents have to communicate with each other, as well as Web services that were built by different organizations. As a result, semantic interoperability difficulties such as polysemy and synonymy can occur. Semantic Web technology solves the polysemy problem by marking up a document or legacy software interface by linking concepts such as classes to other concepts defined in an ontology. The synonymy problem is solved by allowing explicit declarations that term X in an ontology or mark-up is equivalent to term Y in another ontology or mark-up (Kogut and Heflin 2003). Table 17.1 adapted from Sollazzo *et al.* (2002) is helpful in understanding the Semantic Web.

The ontology is the core of the Semantic Web that makes the web machine processable. It covers both vocabularies of concepts and relations and axioms of constraints and rules. Vocabularies of unambiguous domain-specific concepts clarify and share the structure of knowledge. The vocabulary is necessary for geographically distributed development units of knowledge bases and long-term development of knowledge bases such as developing the Semantic Web and its contents. Many definitions of ontology have been produced (Gómez-Pérez 2004), however, the one preferred by the authors is:

An ontology is a formal, explicit specification of a shared conceptualisation. Conceptualisation refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. Explicit means that the type of concepts used, and the constraints on their use are explicitly defined. Formal refers to the fact that the ontology should be machine readable. Shared reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual, but accepted by a group (Studer et al. 1998).

Table 17.1. Comparison of Web service features

Dimension	Current Web	Semantic Web
Service	Simple	Composed
Requestor	Human	Machine
Provider	Registration	No registration
Broker	Key Player	Facilitator
Service description	Taxonomy	Ontology
Descriptive elements	Closed world	Open world
Data exchange	Syntactic based	Semantics based

To facilitate the concept of the Semantic Web, the DAML program officially began in August 2000. Its goal was to create technologies that would enable software agents to dynamically identify and understand information sources, and to provide interoperability among agents in a semantic manner to benefit intelligence analysts, military planners and services such as fire brigades. The technologies included a language and tools that embed DAML mark-up in Web pages and other information sources in a manner that is transparent and beneficial to users. To measure the achievements of this project, an experiment plan was designed to demonstrate the DAML technologies in the context of satisfying key military needs (Flynn and Dean 2002a). The DAML experiment functionality concept is shown in Figure 17.1, which indicates what components and technologies will be incorporated. One of the components, SONAT (Semantic Operational Net Assessment Tool), is an application to support the experiment. As shown in Figure 17.2, the SONAT business process assembles agents and applications to exploit ontologies, data and knowledge bases and outputs preferred courses of action in order to support operations. The SONAT architecture is shown in Figure 17.3, together with its graphical user interface, information sources, ontologies, agents, applications and their interconnections.

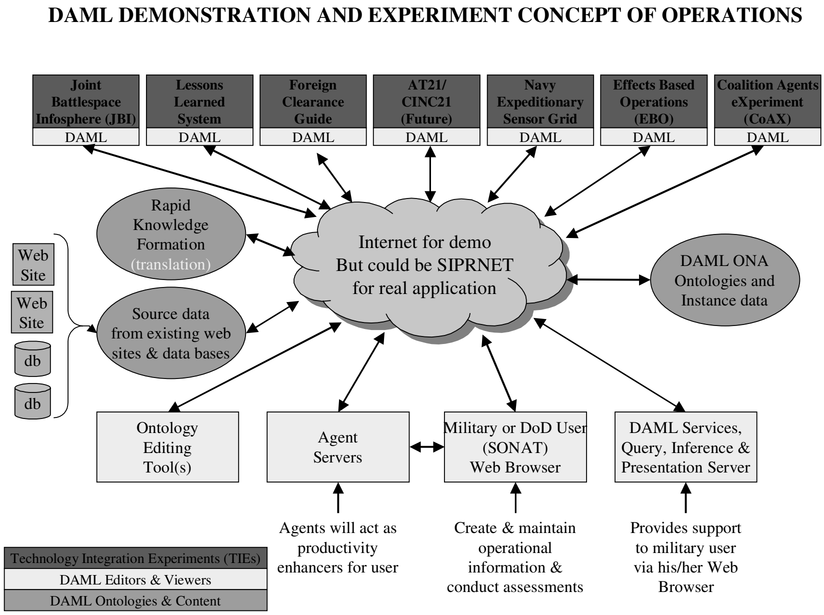


Figure 17.1. DAML experiment functionality concept taken from Flynn and Dean (2002a)

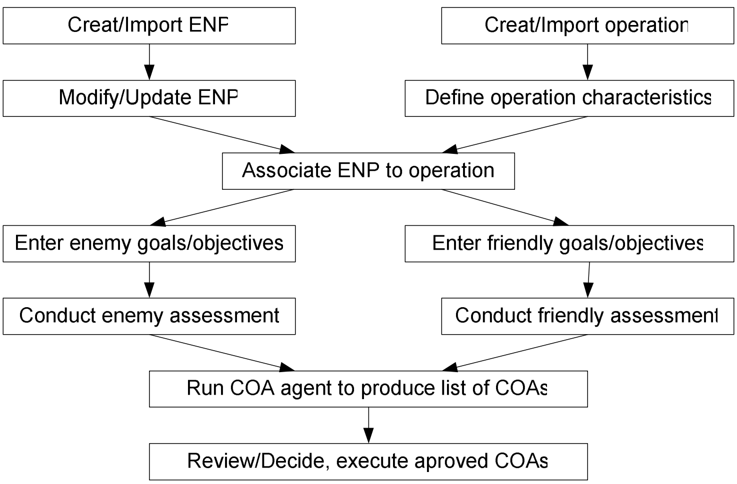


Figure 17.2. SONAT business process taken from Flynn and Dean (2002a) (ENP: elements of national power; COA: course of action)

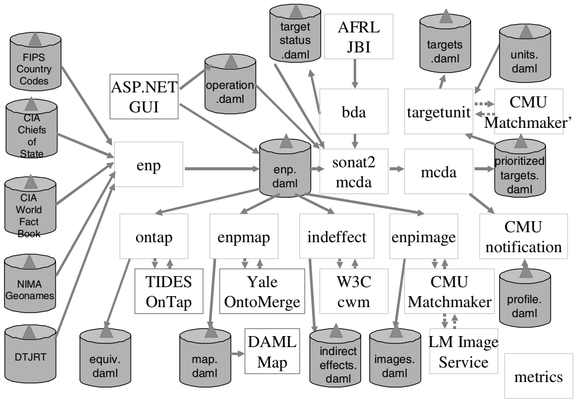


Figure 17.3. SONAT architecture taken from Flynn and Dean (2002b)

The DAML experiment involves many agents as shown in Figure 17.3. The generic architecture for the SONAT agent applications is shown in Figure 17.4. Two of the SONAT agents are multicriteria decision analysis agents based on an analytical hierarchy process, or the COA (course of action) agent, and a sensitivity analysis agent. The COA agent works out weighted values of possible courses of action for attacking enemy vulnerabilities and supporting friendly objectives while the sensitivity analysis agent runs experiments with the stored ontology data to identify the most sensitive parings of ENP (elements of national power) nodes and friendly actions so that relatively small changes in the assigned values could result in significant changes in the overall effects (Flynn and Dean 2002a).

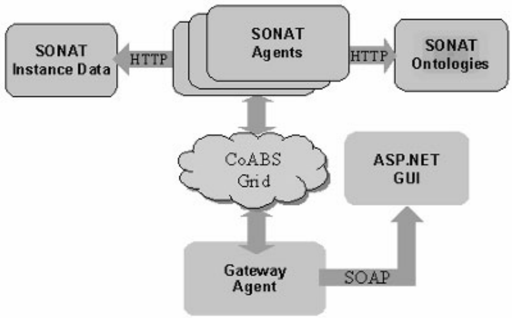


Figure 17.4. Generic architecture for SONAT agent applications taken from Flynn and Dean (2002b)

Wine Agent, developed by Stanford University, is recommended as an example to demonstrate the use of Semantic Web technologies by W3C (2004). It is able to recommend wines that match meal courses using DAML+OIL (ontology inference layer). The example uses a domain ontology to represent foods, wines, their properties and relationships among them. The JTP (Java Theorem Prover) is used to derive the appropriate pairings. In addition, Shadbolt *et al.* (2004) describe an integrated Semantic Web application, that can combine information from multiple heterogeneous sources such as published RDF (Resource Description Framework) sources, personal web pages, and databases in order to provide an integrated view of a multidimensional space and illustrate a number of substantial challenges such as referential integrity, tractable inference and interaction support for the Semantic Web.

The Semantic Web demands many technologies and tools (Flynn and Dean 2002a). In particular, the following key issues are worth considering for developers of the Semantic Web:

- Logic-based Web ontology languages that mark-up information on the Web, and description logic that represents a wide range of knowledge;

- Development tools that edit, browse and manage Semantic Web ontologies;

- Semantic Web ontologies that conceptualize domain knowledge into classes, properties, and relationships between classes and ontology engineering (Gómez-Pérez 2004);

- Intelligent software agents that identify, interpret, manipulate, and interoperate the marked information on the Web across agents and their development tools (Luck *et al.* 2004).

The OWL and Protégé software are much concerned with developing Semantic Web ontologies. The OWL, a description logic-based Web ontology language, which is designed for use by applications that need to process meaning and content of information, was released with W3C recommendations to support advanced Web search, software agents, and knowledge management (W3C, 2004). Thus, OWL can be used to code the Semantic Web ontology directly, or to create development tools for the Semantic Web ontology. For dynamic ontology, the existing description logic is being extended with notions of time (both quantitatively and qualitatively), and with the ability to represent and reason about service actions.

Protégé is an ontology and knowledge-base editor and its OWL Plug-in provides support for editing Semantic Web ontologies in three OWL sublanguages: OWL Lite, OWL DL, and OWL Full. It is suitable for not only system developers but also domain experts. Protégé has the following features (Protégé 2005):

- It is an open-source, Java tool that provides an extensible architecture for the creation of customized knowledge-based applications. Racer is a reasoner and can be used for consistency checking (testing whether a class could have instances) and classification (inferring whether a class is a subclass of another class). OWLViz is a visualization plug-in and enables the class hierarchies in an OWL ontology to be incrementally viewed. After having been plugged into Protégé, Racer and OWLViz enhance Protégé's usability and functionality;

- It can be extended with graphical widgets for tables, diagrams and animation components to access other knowledge-based systems embedded applications;

- Protégé was recommended as a tool to edit Semantic Web ontologies by W3C (2005).

Together with ontologies, software agents underlie the Semantic Web. Once an infrastructure for Semantic Web ontologies has been established, a large number of agents are needed to enable the ontologies, as shown in Figure 17.3. Software agents are a class of software. The user of a software agent must delegate a task to the agent and the latter autonomously carries out the task on behalf of the user. In order to do that, an agent must be able to communicate with other agents and/or its user to receive messages and provide the result of its activities; it must also be able to monitor its execution environment for new messages (Luck *et al.* 2004). Usually an agent-building tool is employed to develop this sophisticated class of software to reduce the complexity, time, and number of errors. In Figure 17.5, an agent and a GUI interface were developed using the AgentBuild of Acronymics, Inc. Each time the Say Hello button is clicked on, a message is sent to the agent and the latter in turn returns a string of "Hello World from Jinwei LU! The time is…" followed by a current time. The message appears in the text field of the GUI. If the Quit button is clicked on the GUI will disappear and the agent engine will shutdown.

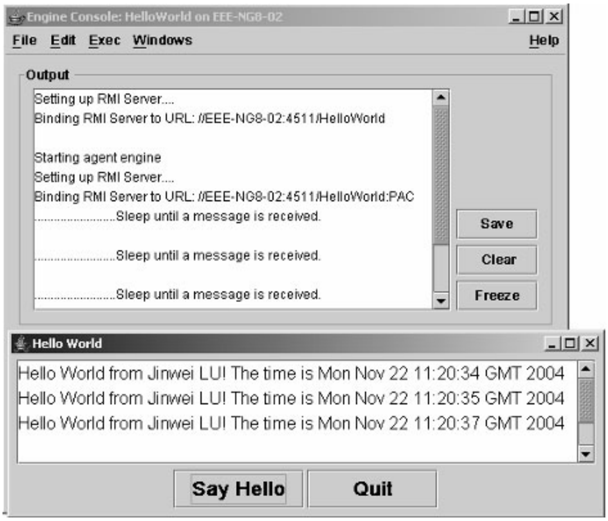


Figure 17.5. An agent and a GUI interface

17.2.2 Benefits of the Semantic Web in the Context of DSS

The benefits of the Semantic Web to DSSs are broad. In establishing requirements for OWL, important application cases of Semantic Web ontologies were considered, including Web portals, multimedia collections, corporate Web site management, design documentation, agents and services and ubiquitous computing (W3C, 2004). Bell *et al.* (2005) have recognized that the concept of ''Semantic Web'' has excited researchers in areas ranging from distributed information systems to artificial intelligence and thus edited a special issue about the Semantic Web and Semantic Web services. On the assumption that e-Commerce using the Semantic Web will be more sophisticated, the integration of Web services, agent cooperation, domain ontologies and data mark-up languages such as OWL, Jutla *et al.* (2005) have proposed a privacy management architecture consisting of client-side and web-side architectural data components and services to assure users of the proper dealings with private data. The availability of Protégé has led to a great increase in applications of Semantic Web ontologies. Rubin *et al.* (2004) represent knowledge in ontology and provide reasoning services for predicting organ injury and its physiological consequences in order to classify injuries and support decisions about diagnosis, triage and additional tests. According to a belief that information retrieval has become a major challenge due to the fact that the volume of on-line information dramatically grows, an information retrieval system has been researched for information discovery in complex and dynamic domains based on the ontology (Currie and Parmadee 2004). Hughes and Crichton (2004) indicated benefits of planetary science ontology to define, classify, validate, data mine, describe, correlative search, and simple search. A project called ArtEquAKT (Alani *et al.* 2004) aims to automatically extract and then feed online information to an ontology, and furthermore generate stories by extracting and structuring information from the knowledge base. Chan *et al.* (2002) developed an ontology for structuring the knowledge base of an expert system.

Several problems in operational railway management identified in Section 17.1 could be approached by incorporating Semantic Web technologies with a railway DSS, which may lead to a railway SONAT counterpart. Resolving these problems gives rise to challenges to knowledge-base development for a large scale system, knowledge management, intelligent knowledge inference and distributed querying in the railway domain. Emerging Semantic Web technologies are effective solutions for these challenges.

17.3 Ontologies for an Internet-based Railway DSS

A Semantic Web ontology for vehicle and track system integration is presented to show functions of inference, visualization, query, *etc.* In addition, the development of a Semantic Web ontology for UK mandatory standards for rail vehicles using Protégé will be briefly described. In order to achieve this, a framework for conceptualization of the rail vehicle/track integration is firstly described.

17.3.1 Framework for Conceptualization in the Domain of Rail Vehicle/Track Interaction

A framework is proposed to identify a large number of knowledge items for railway vehicle/track integration and to organize them in an object-oriented manner in view of structural hierarchy and life-cycle as in Table 17.2. This framework takes advantage of the object specification and is in line with the systems engineering principle of life-cycle considerations. It can be used to identify and organize domain specific knowledge in a structured manner.

Table 17.2. Framework of knowledge identification and organization for rail vehicle/track integration

Structural hierarchy	Aspects in life-cycle
Parts such as wheel and rail	• Design o Geometry o Material and treatment • Manufacturing involving material/metallurgy o Wrought o Cast • Maintenance
Components/subsystems such as wheelset, bogie, track, and vehicle	• Design of structure o Analysis for track profile, wheelset curving theory, and vehicle suspension design and dynamics • Assembly o Process o Measurement • Maintenance
System of vehicle/track	• Design of structure o Analysis for interaction between vehicle and track including contact mechanics, tribology, effects of vehicle and track geometry, *etc.* • Installation • Operation • Maintenance • Reuse

17.3.2 A Railway Ontology for Vehicle and Track System Integration Using Protégé

A Semantic Web ontology for a railway vehicle and track system built using Protégé's OWL Plug-in will be used to demonstrate features such as inference, visualization, and querying. Initially, the property of *affects* was defined to be transitive. Then five classes of *Wheel, Bogie_part_that_affects_Wheel, Rail, Bogie_part_that_affects_Rail*, and *Railnetwork_part_that_affects_Ballast* were defined as in Figure 17.6, where the three definitions for *Bogie_part_that_affects_Wheel, Bogie_part_that_affects_Rail*, and *Railnetwork_part_that_affects_Ballast* are necessary and sufficient. Thus, an asserted hierarchy of structural ontology of rail vehicle/track was built, as in the left-hand side of Figure 17.7, and then Protégé was able to infer a hierarchy, as in the right-hand side, by invoking the Racer reasoner. The inferred hierarchy indicates that the two classes of *Wheel* and *Bogie_part_that_affects_Wheel* are inferred as subclasses of the class of *Bogie_part_that_affects_Rail* due to the *Wheel* definition and the transitive property of *affects* respectively. For the same reasons, the two classes of *Rail* and *Bogie_part_that_affects_Rail* are inferred as subclasses of the class of *Railnetwork_part_that_affects_Ballast*. The inferred conditions for the class of *Wheel* are shown in Figure 17.8. In addition, the asserted conditions of the *Wheel* class in Figure 17.6 also imply that *Wheel* affects *Rail* through the *Force* class. Visualization of class hierarchies in Figure 17.9 was produced for improving data presentation using OWLViz, and can be incrementally navigated.

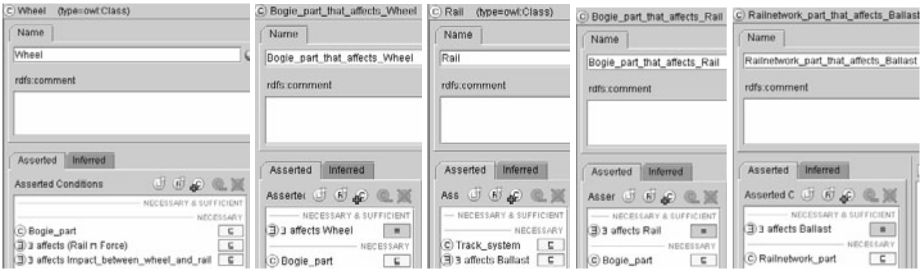


Figure 17.6. Definitions of the five classes of Wheel, Bogie_part_that_affects_Wheel, Rail, Bogie_part_that_affects_Rail, and Railnetwork_part_that_affects_Ballast

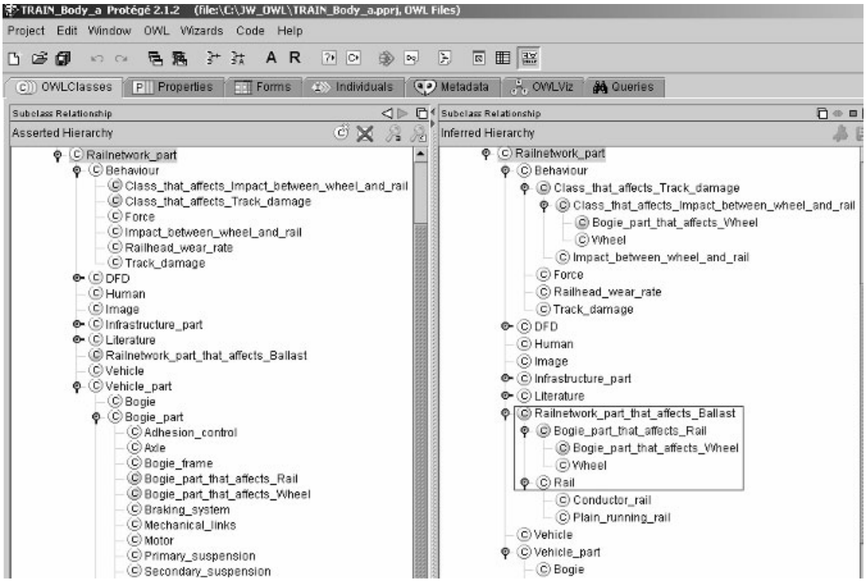


Figure 17.7. Asserted and inferred hierarchies of a structural ontology of the rail network

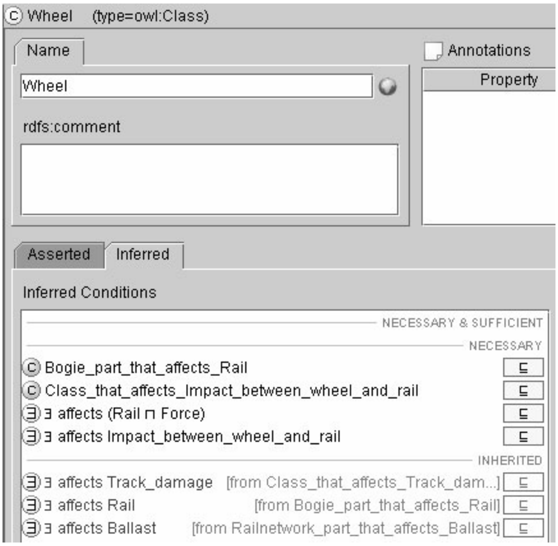


Figure 17.8. Inferred conditions for the Wheel class

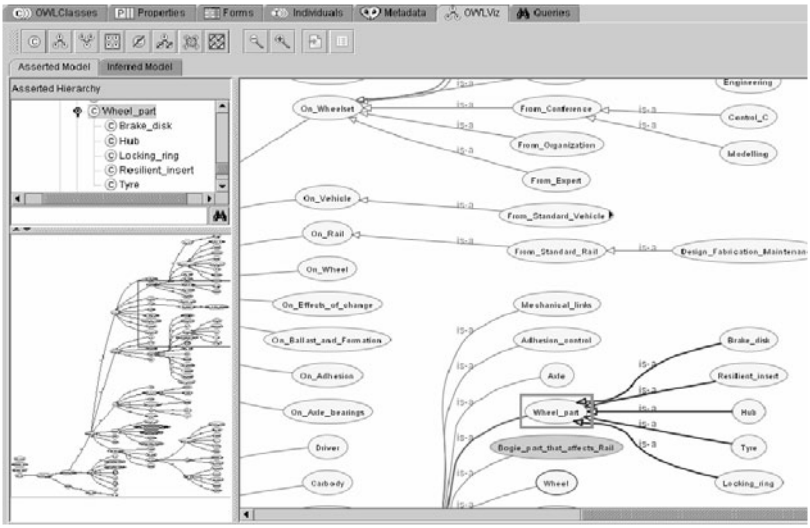


Figure 17.9. Visualization of class hierarchies for improving data presentation

Figure 17.10 shows a literature ontology and some of its instances. These instances are examples of the corresponding classes in forms of standards, journal papers, projects and conference papers, all of which can be queried. Example queries are shown in Figure 17.11.

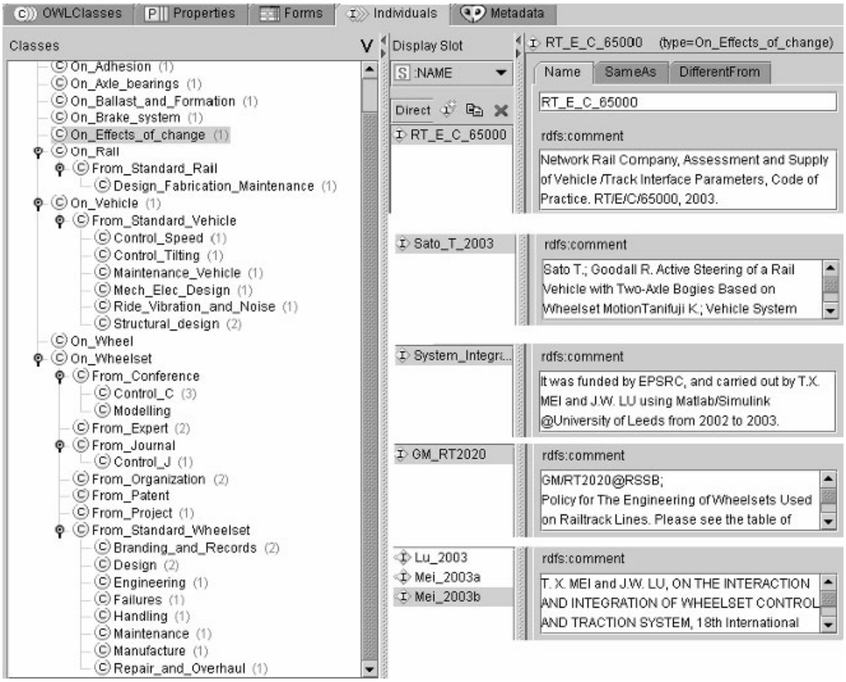


Figure 17.10. Literature ontology and its instances of the rail vehicle/track

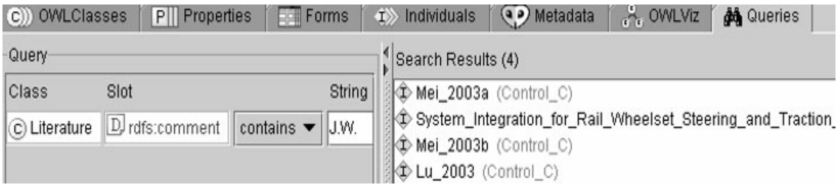


Figure 17.11. Query for items about J.W. from literature instances

17.3.3 An Ontology for UK Mandatory Standards of Rail Vehicles Using Protégé

This section is intended to show a complete knowledge base or a Semantic Web ontology with its complete instances of UK mandatory standards of rail vehicles using Protégé. The UK's Railway Safety and Standards Board have a large index of standards, 58 of which Railway Group Standards for Trains & Rolling Stock are mandatory. Thus, any stakeholder in the UK railway industry has a huge number of related standards to consider, and needs a corresponding ontology to manage those standards. Therefore, a complete ontology was developed for UK mandatory standards of rail vehicles by means of the framework using Protégé. Here, three main activities of conceptualizing the domain, formalizing the domain and implementing the ontology (Gómez-Pérez 2004) will briefly be described.

17.3.3.1 Conceptualization and Formalisation of the Domain
Initially Railway Group Standards on Trains & Rolling Stock were analyzed to identify instances of systems, components and elements together with the aspects of each entity the standards deal with in respect to the framework. The results were then analyzed and formalized into Table 17.3.

Table 17.3. Formalization of Railway Group Standards of Trains & Rolling Stock

Entity	Aspect	No.	Title
Vehicle	Quality	GM/RT2000	Engineering Acceptance of Rail Vehicles
		GM/RT2001	Design Scrutiny for the Acceptance of Rail Vehicles
		…	
	Design	GM/RT2100	Structural Requirements for Railway Vehicles
		GM/RT2120	Requirements for the Control of Risks Arising from Fires on Railway Vehicles
		…	
	Maintenance	GM/RT2004	Requirements for Rail Vehicle Maintenance
		GM/RT2455	Freight Vehicle In-Service Inspections
	Test	GM/RT2273	Post Incident and Post Accident Testing of Vehicles
Air quality and lighting environment	Design	GM/RT2176	Air Quality and Lighting Environment for Traincrew Inside Railway Vehicles
Brakeing system	Design	GM/RT2040	Calculation of Brake Force Data for Rolling Stock Library
		GM/RT2041	Braking System Requirements and Performance for Trailer Coaching Stock
		…	
…			

17.3.3.2 Implementation
Protégé was used to build a hierarchy of classes from Table 17.3. Instances of these standards were named and built as instances of corresponding classes as shown in Figure 17.12. As a result, a complete knowledge base about UK mandatory standards of rail vehicles was established as an element of a more comprehensive KB. The KB is useful to classify existing standards and identify where gaps exist.

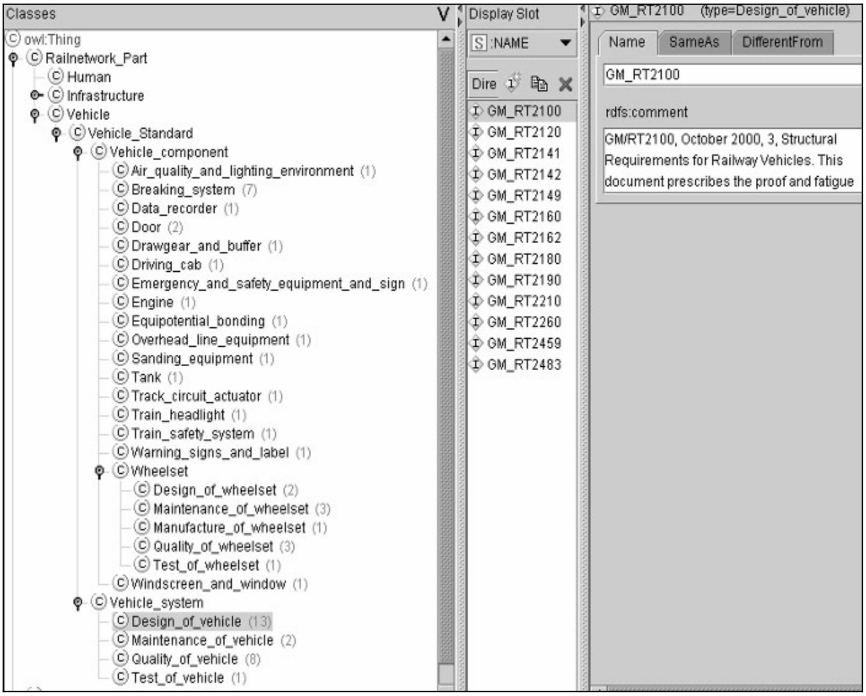


Figure 17.12. Ontology for UK mandatory standards of Trains & Rolling Stock

17.4 Summary

In this chapter, the fundamentals and benefits of the Semantic Web have been introduced through the use of examples, the DAML project, SONAT architectures and OWL and Protégé software.

A Semantic Web ontology for vehicle and track system integration was exhibited to show its capabilities of inference, visualization, query, *etc.* A complete knowledge base or a Semantic Web ontology with all instances for UK mandatory standards of rail vehicles using Protégé was described. These can be used as stand alone components or made available over the Internet as a part of the Semantic Web. This latter scenario would allow them to be interpreted by intelligent software agents. For these ontologies, a framework for conceptualizing the domain of rail vehicle/track was also proposed.

The chapter also indicated how techniques such as Semantic Web ontology and intelligent software agents can be effectively integrated with decision support techniques such as the analytical hierarchy process and sensitivity analysis to improve the performance and management of complex real-world systems like the railways.

In is anticipated that in the future the Semantic Web will become a very useful tool for the decision support community. The DAML project has created Semantic

Web technologies and associated applications such as SONAT that can be used as an example for the development of new intelligent decision support systems. The release of the OWL standard and the availability of Semantic Web ontology tools such as Protégé will greatly promote widespread use of Semantic Web ontologies for decision support. In turn, the population of ontologies will provide an infrastructure with further calls for more intelligent software agents to automate the Semantic Web. Many agent-oriented tools have appeared recently (Luck *et al.* 2004), all of these provide a great opportunity for research and development in the area of intelligent decision support systems.

References

Alani H, Kim S, Millard DE, Weal MJ, Hall W, Lewis PH, Shadbolt N (2004) Using protégé for automatic ontology instantiation. 7th International Protégé Conference, USA

Bayati H, Iwnicki SD, Stow JM (2002) Opportunities for improving interfaces between railway engineering analysis tools. Railway Engineering 2002, London

Bell D, Bussler C, Yang J (eds.) (2005) The semantic web and web services. Information Systems, 30(2) (editorial preface)

Berners-Lee T, Hendler J, Lassila O (2001) The Semantic Web, a new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. The Scientific American, 284(5): 34-43

Chan CW, Peng Y, Chen LL (2002) Knowledge acquisition and ontology modelling for construction of a control and monitoring expert system. International Journal of Systems Science 33(6):485-503

Currie E, Parmelee M (2004) Toward a knowledge-based solution for information discovery in complex and dynamic domains. 7th International Protégé Conference, USA

Flynn J, Dean M (2002a) DAML Experiment Plan. https://www.daml.org/2002/01/experiment/daml_plan_point8.doc

Flynn J, Dean M (2002b) DAML Experiment---Semantic Operational Net Assessment Tool (SONAT). Status Report

Gómez-Pérez A (2004) Ontological engineering: with examples from the areas of knowledge management. Springer, London

Hendler J, Berners-Lee T, Miller E (2002) Integrating applications on the semantic web. Journal of the Institute of Electrical Engineers of Japan 122(10): 676-680.

Hughes JS, Crichton D (2004) The planetary science ontology: a case study in an ontology-based information architecture. 7th International Protégé Conference, USA

Kogut P, Heflin J (2003) Semantic Web technologies for aerospace. IEEE Aerospace Conference of Big Sky: 6_2887-6_2894

Jutla DN, Bodorik P, Zhang Y (2005) PeCAN: an architecture for users' privacy-aware electronic commerce contexts on the semantic web. Information Systems, 30(2): n.a.

Luck M, Ashri R, D'Invern M (2004) Agent-based software development. Artech House, London

Protégé (2005) http://protege.stanford.edu

Rubin DL, Bashir Y, Grossman D, Dev P, Musen MA (2004) Integrating ontologies with three-dimensional models of anatomy. 7th International Protégé Conference, USA

Shadbolt NR, Gibbins N, Glaser H, Harris S, Schraefel MC (2004) Walking through CS AKTive space: a demonstration of an integrated Semantic Web application. Journal of Web Semantics 1(4): 415-419

Shim JP, Warkentin M, Courtney JF, Power DJ, Sharda R, Carlsson C (2002) Past, present, and future of decision support technology. Decision Support Systems 33:111–126

Simon HA, Associates (1986) Decision-making and problem solving. National Academy Press, USA
Sollazzo T, Handschuh S, Staab S, Frank M (2002) Semantic Web service architecture -- evolving web service standards toward the Semantic Web. 15th International FLAIRS Conference, Pensacola Beach, Florida, May 16-18
Stribersky A, Moser F, Peters J, Schandl G (2004) Innovative system dynamics analyses of rail vehicles. Vehicle System Dynamics 41 suppl: 448-457
Studer R, Benjamins R, Fensel D (1998) Knowledge engineering: principles and methods. Data & Knowledge Engineering 25: 161-197
Turban E (1995) Decision support and expert systems: management support systems, 4th edn. Prentice-Hall International, London
W3C (2004) http://www.w3.org

Part III

Trends of Intelligent Decision-making Support Systems

18

The Challenge of Supporting Emerging Inference-based Decision-making

C. Gustav Lundberg

A.J. Palumbo School of Business Administration & John F. Donahue Graduate School of Business, Duquesne University, 600 Forbes Avenue, Pittsburgh, PA 15282, USA,
e-mail: lundberg@duq.edu, and
Swedish School of Economics and Business Administration, Helsinki, Finland

This chapter addresses the complexities of inference-based decision making and the challenge of supporting such processes. People are often unaware of the cognitive framework that facilitates and biases complex decisions. Inference-based decision making is described as bidirectional reasoning. We show how the decision maker gradually makes sense of and simplifies the decision task, how decision criteria can be modeled, and how criteria change as the decision-maker's experience increases. It is claimed that the prime way to aid inference-based decision making is to make the process of generating sense explicit, and that experiments constitute an important tool.

18.1 Introduction

During the relatively short lifespan of decision-making support system (DMSS) research and application, great strides have been made both on the organizational and the technical fronts. An impressive and diverse array of models has been built with the goal of aiding professional decision making in an ever-expanding number of domains. However, one can argue that relatively little research has focused on *why* and *in which circumstances* humans need decision support. In many cases we can sense a "build it and they will come" mentality. This partly accounts for the relatively large failure rate of DMSSs.

In this chapter the processes associated with emerging inference-based decision making are addressed. When we infer, we conclude something on the basis of evidence or reasoning, or even make a reasonable guess. Many important and consequential decisions can be viewed as a gradual process of making sense of a complex decision task. Thus, the focus here is on emerging inference-based decision making, and on the challenges and opportunities associated with attempts to support such intricate processes. People have been shown to generate sense quite effortlessly - and compulsively - even in highly opaque and conflict-laden situations based on multitudes of inferences. However, the process of making sense is poorly understood. Here, an attempt is made to describe the underlying cognitive processes, and to suggest avenues for decision support.

The process of decision making has been described in various ways. Simon (1960) divides the process into three major elements: (1) finding occasions for making a decision, (2) finding courses of action, and (3) choosing between courses of action. More specifically, the predecision process involves the setting of objectives and criteria, information gathering, and the development and evaluation of alternatives. Many researchers, among them Janis (1968), also include a post-decision phase: commitment and adherence to choice, implementation of decision, and/or follow-up and control.

A growing literature is focused on motivated reasoning; on whether individuals distort *information during* the process of reaching a decision, when *justifying* the decision, and/or the *decision criteria* they use when making and/or justifying the decision (cf. Kunda 1990, Brownstein 2003, Phillips 2002, Davis and Ashton, 2002). Kunda (1990) and others have argued that self-serving distortions potentially influence virtually all elements of decision-making, notably problem definition and representation, and information retrieval and recall. In Slovic's (1995) terms, *e. g.* people are involved in the process of constructing preferences (*cf.* Payne *et al.* 1993, Simon *et al.* 2004). Distortion of information and criteria during the process of reaching a decision may result in biased or even faulty decisions, whereas distortion of information and criteria after the decision may influence similar decisions made in the future, and thus lead to faulty or impeded learning. Postdecision distortion makes decisions appear well justified, and, thus, may make decision makers feel unduly confident and may deter them from reflecting on their reasoning (Phillips 2002; Sá *et al.* 1999). Phillips notes that postdecisional distortion for a current decision becomes predecisional distortion for subsequent decision making. Svenson (1998) points out that a decision-maker's representation of a decision problem depends on the context, the preliminary choice alternative, and uncertain and contextually dependent criterion values. This process often becomes biased in support of a preliminary or final choice alternative (*cf.* Montgomery 1983).

Recently, Holyoak and Simon (1999) have suggested that the making of an inference-based decision may be accompanied by an increase in the coherence of assessments of the individual arguments related to the alternatives at hand. In these emerging decisions, assessments of inferences increasingly spread apart, with those supporting the favored alternative growing stronger, and those assessments supporting the less-favored alternative growing weaker. This suggests a bidirectional (constraint satisfaction) reasoning process where *the decisions are based on the inferences made from the provided information, and the emerging decisions, in turn, work backward to alter the strength of the inferences in order to yield coherent support*. This process may continue until sufficient differentiation (*cf.* Svenson, 1992, 1999) has been generated between the competing beliefs, and the decision maker is able to make a decision. It is likely that this bifurcation process is affected by biases, memory of similar instances, analogies, and professional norms. For example, auditors have been shown to carry a positive bias with respect to audit report choice. This initial bias may then facilitate the generation of sense and lead to the construction of an active context (Brézillon *et al.* 2002) by creating increasingly strong links and relationships between decision variables. However, this process is likely to be nonmonotonic; possibly characterized by abduction: the promotion of the best fitting theory (Lundberg 2000). The initial bias toward a particular alternative may also be rooted in affect (*cf.* Slovic *et al.* 2002).

More generally, making sense can be viewed as the activity of fitting something confusing into coherent mental representations that include concepts, beliefs, goals, and actions. Coherence theory (Thagard 2001) approaches problems in terms of the satisfaction of multiple interacting soft constraints with highly interconnected elements. The elements are propositional or other types of representations that are connected via weighted links of coherence (positive or excitatory constraints) and incoherence (negative, interfering, or inhibitory constraints). Connection weights represent the sign and strength of the relations and are bidirectional to permit cognitions to mutually influence each other. External inputs to units represent influences from the environment, while internal constraints (*cf.* Shultz *et al.* 2001) involve relations among the elements. Parallel constraint satisfaction is then achieved by algorithms for updating the activations of interlinked units. For example, Holyoak and Thagard's (1997) multiconstraint analogy theory postulates three basic constraints: similarity, structure, and purpose. People's use of analogy is guided by these constraints whose constant interplay encourages coherence; the resolution of local contradictions between constraints and the movement toward a satisfying (internally coherent) compromise. In such spreading activation networks, coherence is a state where similarly implicated inferences are similarly activated.

18.2 Experimentation as a Means to Make Implicit Decision-making Processes Explicit

We have argued that people are compulsive sense makers who are largely unaware of the underlying cognitive processes; unaware of both the ability human decision makers show in integrating complex and often conflicting information, and the biases and fallacies that may influence inference-based decision-making. As it most often is impossible to directly aid the frequent and diverse decision processes as they unfold, we suggest that the process more generally can be aided by making it explicit, thus heightening the decision-maker's awareness and vigilance. As in the cases of judgment biases and decision-making heuristics, people are often unaware of the essence of the cognitive framework that facilitates and concurrently biases their decision. If decision makers are made aware of the underlying cognitive processes, they may also be able to withstand some of the biasing forces.

In what follows, the results of a set of naturalistic experiments are presented. The experiments address both inference-based decision making assumed to require no specialized knowledge, in the form of simulated juries (*cf.* Lundberg 2004), and inference-based decision making that requires expertise, auditors making going concern judgments (*cf.* Lundberg and Nagle 2002, 2005). In both cases the experiments are designed to be maximally naturalistic. However, even if the experiments touch upon both expert and common-sense decision making, a review of the expertise literature is beyond the scope of this chapter. For interesting overviews please consult Andersson (2004), Sniezek *et al.* (2004), Bolger and Wright (1994), and Shanteau and Stewart (1992).

Andersson (2001, 2004) utilizes Web-based software to trace the decision processes of experienced loan officers and novices. The participants in this simulation interacted with the software and decided whether a loan to a small firm should be granted or rejected. The software includes, in principle, all the information that is available for Swedish loan officers when making a decision about a small business loan: 74 cues belonging to four categories of information. All cues were hidden and had to be requested one by one. If a loan was granted, the participant stated the interest rate and years of amortization, and provided an explanation. If a loan was denied, the participant stated why the application was rejected and what further information was required. Andersson (2004) found that experts searched for significantly more cues than inexperienced participants did, but also that the experts did not reveal a high degree of agreement (consistency) in their judgments. These process insights would not have been very easy to obtain without the complete, naturalistic, and interactive software; and thus without an experimental framework. It is likely that the insights into the decision process would be of interest both to the experts themselves and to their organizations in their respective drives to improve the decisions and the decision consistency.

Next, two experiments are discussed in some detail, followed by a set of hypotheses, and a sample of the results. Like the Andersson (2004) findings, our results may be of interest both to individuals trying to improve their decisions by understanding the facilitating and biasing processes and to groups of decision makers and organizations.

18.2.1 Experiment 1

Experiment 1 is set in external auditing and thus requires domain knowledge. During the course of an audit engagement an auditor is responsible for predicting whether the client has the ability to continue as a going concern for a reasonable period of time (generally one year). In cases where an auditor has substantial doubt about a company's ability to continue as a viable entity, s/he may modify the standard *unmodified audit report*, UNM, to indicate these concerns (a *going-concern modification,* GCM). Asare (1992) divides the going-concern modification process into two phases. An auditor first collects and evaluates evidence, E, in order to reach a subjective belief, $P(C|E)$, where C is the firm's continued existence. The auditor then compares $P(C|E)$ to the threshold $P^*(C)$ at which the auditor will have *substantial doubt* about the entity's ability to continue as a going concern.

The participants were 127 professional auditors on all levels of expertise, from beginner staff auditors to partners (18 partners, 37 managers, and 72 staff auditors). The participants had an average of 7.6 years of experience in external auditing, and represented 13 auditing firms of all sizes in the Pittsburgh, Pennsylvania area and in Cincinnati and Columbus, Ohio. The experiments are based on a comprehensive (2643-word) instrument containing an actual business entity's financial statements, selected financial statement notes, and other relevant company-specific and industry-related information. The company upon which the instrument is based received an unmodified audit report from its auditors, but subsequently filed for bankruptcy within the next year. We excluded the auditor's report choice, changed the company name (to disguise its identity), and, in order to save the participants' time, removed a number of verbal management notes that a small panel of experts labeled neutral (*cf.* Lundberg and Nagle, 2002).

The auditors were first presented with short scenarios (depicting nine key aspects: *Cash Flow*, *Liquidity*, *Capital Structure*, *Profitability*, *Revenue*, *Financial Flexibility*, *Industry Trend*, *Operational Structure*, and *Litigation*), all originating in the instrument. The selection was based on the signals that 55 professional auditors in an earlier experiment had isolated as the most crucial signals influencing their report choice (Lundberg and Nagle 2002). The scenarios were first presented as isolated cases. The participants were asked to rate each aspect as to what type of report it would support, and how strongly, on a continuous scale ranging from *Very strong support* for a going concern modification (GCM) (0) over *Neutral* (62.5) to *Very strong support* for an unmodified audit report (UNM) (125). Next, the participants were introduced to the comprehensive instrument itself –and thus to the context– and asked to identify prominent positive (supporting an UNM) and negative (supporting a GCM) signals.

After 15 minutes the participants were asked *not* to make a report decision, but to indicate on a continuous scale (ranging from 0 to 125) which way they were leaning: toward a GCM or toward an UNM, and to rate (same scale as previous aspect ratings) how much support each of the nine key aspects provided for or against their leaning.

Upon completion of the instrument analysis, the participants were asked to make a report decision and again to rate the nine aspects. Following a break of approximately 25 minutes, each participant was asked to *replicate* their decision phase aspect strength ratings. Thus, the experiment consists of four phases: *Pre*, *Leaning*, *Decision*, and *Post*.

18.2.2 Experiment 2

Experiment 2 takes the form of juror decision making and thus is based largely on commonsense reasoning. Here, the participants consider the negligence or lack of negligence of an auditor. All 47 participants were graduate (mostly business) students in an *Information Ethics and Legal Issues* course at two Pennsylvania universities. The participant pool was considered suitable for the experiment as all the participants were potential jurors. The average age of the participants was 32.

The case, based on a scenario developed by Lowe and Recker (1994), describes Atlantis Corporation, a producer of electronic toys, in a market characterized as very unstable. Increased competition has eroded Atlantis' market share, cash flow, and profit. Atlantis has been audited for years X-3 and X-2 by a large international auditing firm that issued (standard) favorable audit reports. During the year X-1 audit, the financial information was considered to be slightly below industry averages. The auditing firm's standard audit procedures revealed certain conditions concerning Atlantis' financial viability: *Safety Standards* (negative), *Labor Negotiations* (negative), *Toy Industry* (positive), *Cash Problems of Biggest Customer* (negative), *Relationship with Suppliers* (positive), and *Patent* (positive).

The experiment participants were first asked to review the information and to indicate to what extent they thought each of the six conditions supports either type (UNM or GCM) of audit report. The continuous rating scale ranged from *Strong*

Support for a Non-favorable Audit Report (-5) over *Neutral* (0) to *Strong Support for a Favorable Audit Report* (+5).

The participants were then informed that the auditing firm had issued a standard, favorable audit report without detailing concerns for Atlantis' continued existence. The participants were also informed that: (i) the company's biggest customer declared bankruptcy during the early part of April of year X, (ii) labor began their indefinite strike on June 1, year X, (iii) Congress (in August) passed stricter toy safety standards forcing Atlantis to stop making several products that were sold as scrap, and (iv) that Atlantis filed for bankruptcy in September, year X. Shortly thereafter, stockholders who had bought stock during the past year filed suit against the auditing firm, claiming that the auditing firm incorrectly provided a favorable audit report and that the public should have been informed about important disclosures of uncertainties that led to the bankruptcy.

The experiment participants were next asked to assume that they had been selected for jury duty. They were informed that they were in the empanelled phase of the process and instructed to listen to the arguments of the prosecutor and the defense attorneys, but *not* to make a judgment. A short summary containing the prosecutor's claim of the auditing firms' negligence and the defense's counterclaim was provided. The defense warned the jurors about the perils of hindsight biases. The participants were then asked to once more review the case and the six key conditions. After completing the review, the participants were reminded of the advice not to make a judgment at this stage, yet asked to indicate which way they were leaning - toward finding the auditing firm either *negligent* or *not negligent* - and to indicate to what extent they thought that each of the six conditions supported their *leaning*.

Finally, in the deliberation phase participants were introduced to the two verdict options: *to find the auditing firm guilty of negligence* or *to find the auditing firm not guilty of negligence*. They were told to get prepared to vote, and asked to review the data and the arguments once more. Upon finishing the review, they were asked to make their decision (negligent/not negligent), and to indicate to what extent they thought each of the six conditions supported their *verdict*. Thus, the case consisted of three stages: *Pre*, *Leaning*, and *Decision (Verdict)*.

18.2.3 Hypotheses and Propositions

In accordance with the predictions originating in coherence theory and multiple-constraint satisfaction modeling, the following three hypotheses and one proposition are forwarded:

Hypothesis 1: The assessments of inferences will increasingly spread apart as the participants progress through the decision-making stages, with those supporting the favored alternative growing stronger, and those supporting the less-favored alternative growing weaker.

Hypothesis 2: The correlations between the dependent variable (the decision) and the independent variables (aspects) will gradually grow stronger and increasingly take on the same sign (as will those between the aspects) as the participants progress through the decision stages.

Hypothesis 3: The decision-maker's mental model of the task will gradually become simpler as s/he progresses through the decision stages (Lundberg 2005).

Proposition 1: The decision criteria are likely to vary with the decision-maker's experience. More specifically, the more experience an auditor has, the more risk s/he is willing to take (*cf.* Lundberg and Nagle 2005).

Importantly, all through we claim that the decision maker is largely unaware of these processes and how they may aid/bias the decision.

18.3 Results

18.3.1 Alternative Differentiation

We first explore the hypothesis that the assessments of inferences increasingly spread apart, *i. e.* to what extent the decision makers manage to differentiate the alternatives. Importantly, both experiments reveal roughly the same overall tendency. The average aspect ratings over four and three decision stages, respectively, are depicted for auditors who chose an UNM versus those who chose a GCM (Figure 18.1), and jurors who found the auditor not negligent and those who found the auditor negligent (Figure 18.2). In Experiment 1 (Figure 18.1), a rating of 62.5 is perfectly neutral, as supportive of either audit report, whereas the point of neutrality in Experiment 2 (Figure 18.2) is zero.

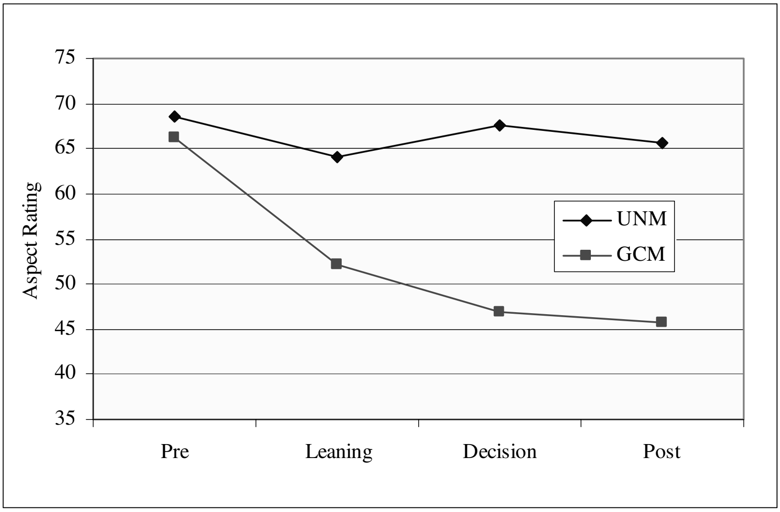


Figure 18.1. Average aspect ratings over four decision stages: Auditors

Figure 18.1 reveals little differentiation in the *Pre* stage, followed by increased differentiation in the *Leaning* and particularly in the *Decision* stage. Thus measured, no further differentiation occurs between the *Decision* and the *Post* stages. There is support for the notion that decision makers isolate a favored alternative early and then build support for that alternative. Even in this complex decision task, 80% of the auditors chose the alternative toward which they earlier were leaning. Of the 25 auditors who changed their report choice between the *Leaning* and *Decision* phases, 15 auditors switched from a GCM leaning to an UNM choice, and 10 from an UNM leaning to a GCM choice.

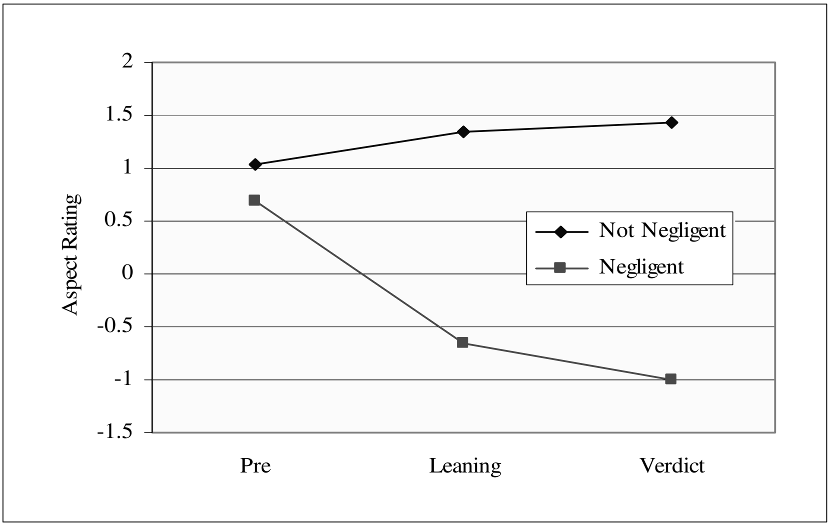


Figure 18.2. Average aspect ratings over three decision stages: Jurors

It is particularly interesting that the differentiation between the alternatives is so strong in Experiment 2 (Figure 18.2), as the information amount is limited to six aspects and the information stays the same in all decision-making stages. The sense that so clearly emerges in each verdict camp therefore must be generated internally, by each juror generating a mental image that better and better fits the intended/final verdict. Also here, most jurors (81%) stuck to the verdict alternative that they earlier reported to lean toward, building further support for their decision as they progressed through the decision stages.

The gradual adjustments to the various (6) aspects in Experiment 2 are depicted in Figure 18.3. Here, the evolution of each aspect's (A1 to A6) ratings is depicted in six columns, three each for the two possible verdicts for the Pre, Leaning, and Verdict stages. In the process of making sense of the information, the jurors who found the auditor not negligent did not bolster the ratings of the clearly positive aspects, A3, A5, and A6. However, A1 that was seen as slightly negative in the Pre phase became positive, and A2 shifted from negative to marginally positive. The clearly negative A4, in turn, became less negative in the Leaning and Verdict phases.

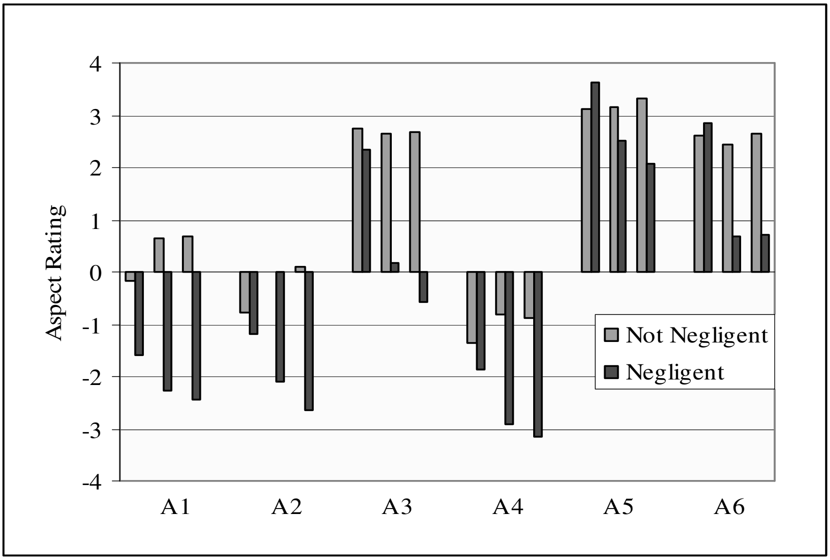


Figure 18.3. Average aspect ratings per decision phase: Jurors

Jurors who found the auditor negligent gradually constructed a very different sense. Here, the aspects that received a negative rating in the Pre phase (A1, A2, and A4) became drastically more negative. The positive aspects A5 and A6 became less positive, whereas A3 went from clearly positive to slightly negative. It is likely that the original aspect ratings generated a preliminary verdict, and that the preliminary verdict in turn influenced the aspect ratings. Importantly, the participants were not aware of this drift in the aspect ratings. This phenomenon also highlights the importance (and inherent danger) of the "stories" both defense and prosecuting attorneys try to weave for the jurors.

18.3.2 Changes in the Correlations Between Decision and Aspects

It was hypothesized that the correlations between the dependent variable (the decision) and the independent variables (aspects) gradually would grow stronger and would take on the same sign (as would those between the aspects) as the participants progress through the decision stages. This would indicate increasing coherence. There is very clear support for this hypothesis. Whereas, in section 18.3.1, we discussed *to what extent* differentiation between alternatives is constructed, here we address *how* differentiation may be constructed. Table 18.1 depicts the structuring that occurs among the auditors making a going concern decision in Experiment 1. In order to facilitate the interpretation of the table, the significance of the correlation coefficients is marked by font size, with the very small numbers indicating a nonsignificant correlation, the next larger size significance for an α of 0.1, and the largest two significance for αs of 0.05 and 0.01 (bold), respectively. The shaded coefficients occurring in the *Pre*, *Leaning*, and *Decision* stages are negative. Please note that full aspect labels are listed in Section 18.2.1.

Table 18.1. Correlations between decision and aspects in Experiment 1

	Dec	CaFl	Liq	CapSt	Prof	Rev	FinFl	IndTr	OpSt
CaFl	0.02	1							
Liq	-0.13	**0.30**	1						**PRE**
CapSt	0.05	**0.27**	**0.50**	1					
Prof	0.01	-0.06	0.16	0.11	1				
Rev	0.08	**0.41**	**0.43**	**0.46**	-0.07	1			
FinFl	0.07	**0.24**	0.10	0.15	0.11	**0.23**	1		
IndTr	0.14	**-0.23**	-0.07	-0.10	**0.36**	**-0.29**	0.10	1	
OpSt	0.06	0.15	**0.24**	**0.37**	-0.11	**0.42**	0.15	-0.06	1
Lit	0.10	-0.06	0.00	0.07	0.11	-0.02	0.09	0.22	0.09
CaFl	**0.34**	1							
Liq	**0.27**	**0.27**	1						**LEAN**
CapSt	**0.24**	**0.40**	**0.37**	1					
Prof	**0.25**	**0.34**	0.21	0.21	1				
Rev	**0.35**	**0.37**	0.21	**0.29**	**0.34**	1			
FinFl	0.14	0.07	**0.30**	0.19	0.22	-0.06	1		
IndTr	**0.25**	**0.25**	**0.27**	**0.33**	**0.43**	**0.36**	0.10	1	
OpSt	**0.33**	**0.34**	0.07	0.13	0.08	**0.37**	0.04	0.12	1
Lit	0.14	0.09	0.05	-0.07	0.21	0.09	0.09	0.19	0.08
CaFl	**0.54**	1							
Liq	**0.46**	**0.57**	1						**DECIS**
CapSt	**0.41**	**0.42**	**0.56**	1					
Prof	**0.45**	**0.38**	**0.52**	**0.58**	1				
Rev	**0.52**	**0.35**	**0.28**	**0.37**	**0.34**	1			
FinFl	**0.43**	**0.45**	**0.37**	**0.26**	**0.26**	0.08	1		
IndTr	**0.44**	**0.26**	**0.24**	**0.38**	**0.46**	**0.47**	**0.34**	1	
OpSt	**0.37**	**0.36**	**0.25**	**0.28**	0.09	**0.35**	**0.28**	**0.41**	1
Lit	0.23	0.04	-0.01	0.03	0.07	0.09	0.17	**0.39**	0.19
CaFl	**0.49**	1							
Liq	**0.44**	**0.60**	1						**POST**
CapSt	**0.40**	**0.40**	**0.53**	1					
Prof	**0.42**	**0.38**	**0.52**	**0.64**	1				
Rev	**0.52**	**0.28**	**0.27**	**0.48**	**0.42**	1			
FinFl	**0.40**	**0.35**	**0.45**	**0.45**	**0.36**	**0.26**	1		
IndTr	**0.40**	**0.30**	**0.29**	**0.39**	**0.45**	**0.36**	**0.35**	1	
OpSt	**0.41**	0.18	0.22	**0.37**	0.16	**0.37**	**0.40**	**0.27**	1
Lit	**0.26**	0.03	0.02	0.11	0.05	0.07	**0.34**	**0.30**	**0.26**

Clearly, the correlations between the dependent and the nine independent variables gradually increase as the auditors progress from stage to stage. In the *Pre* stage, none of the nine correlation coefficients are significant, whereas in the *Leaning* and *Decision* stages, 7 and 8 coefficients, respectively, are significant. When the auditors look back at their decisions (in the *Post* stage), even more sense has been made of the relationships, with all nine coefficients being significant. Thus, despite the fact that Figure 1 revealed no further differentiation of the alternatives in the *Post* stage, the process of information structuring did continue in this phase. Similarly, the correlation coefficients between the nine aspects gradually become stronger and all positive. Of the 36 coefficients, 17 in the *Pre* stage were significant compared to 31 in the *Post* stage. It appears that the cost of increasing sense is multicollinearity.

A similar information structuring pattern emerges in the three decision stages in Experiment 2. In the *Pre* stage, one of the six correlation coefficients between the dependent and independent variables is significant, whereas in the *Leaning* and *Decision* stages, 5 and 6 coefficients, respectively, are significant. Of the 15 coefficients between the six aspects, 5 in the *Pre* stage were significant compared to 9 in the *Leaning* and *Post* stages. All in all, in both the auditor and the juror case, there is evidence of purposeful, albeit largely unconscious, information structuring. Sense is being constructed, and sense facilitates the decision.

18.3.3 Simplification of Mental Model

It was hypothesized that the decision maker would generate an increasingly simple model of the decision task as s/he progresses through the decision stages. The above correlation analyses suggest that significant aspect restructuring occurs as the decision makers progress toward a decision. Factor analysis provides another way to illustrate the information structuring process in multiaspect decision-making settings. Lundberg (2005) found that the number of significant factors decreases as the decision makers progress toward a final decision; as they make more sense of the decision problem at hand. The factor structures associated with Experiments 1 and 2 were explored. The models are based on principal-component analysis and Varimax rotation. A factor was considered significant if its eigenvalue exceeded 1. In the *Pre* stage of Experiment 1, four components are significant, decreasing to three in the *Leaning* and *Decision* stages, and finally to two in the *Post* stage (indicated as shaded cells in Table 18.2). In the *Decision* stage, Component 1 is defined by *Capital Structure*, *Profitability*, and *Revenue*, Component 2 by *Cash Flow*, *Liquidity*, and *Financial Flexibility*, and Component 3 by *Industry Trends*, *Operational Structure*, and *Litigation*. As indicated in Table 18.2, the three factors account for 65.8% of the variance.

The factor structure of the *Post* stage aspect ratings is depicted in Table 18.3. The table reveals a very simple structure consisting of two quite easily interpretable components. Component 1, with high factor loadings for *Cash Flow*, *Liquidity*, *Capital Structure*, *Profitability*, and *Revenue*, can be labeled ‘Cash Maintenance.’ Its origin is in the largely quantitative, tabular summaries of the company’s financial performance over the last three years. Component 2, with high factor loadings for *Financial Flexibility*, *Industry Trends*, *Operational Structure*, and *Litigation*, can be labeled ‘Structure Maintenance.’ The two components also have a distinct internal (C1) – external (C2) dimension. The two components account for 55.9% of the variance (cf. Table 18.2). Again, the further simplification of the factor structure between the decision and the postdecision stages indicate that the information restructuring continues past the decision itself.

Table 18.2. Per cent of variance explained in auditor models

Stage	2 Factors	3 Factors	4 Factors
Pre	46.68	58.12	69.05
Leaning	45.01	57.83	68.08
Decision	54.55	65.8	75.98
Post	55.93	66.2	75.22

Table 18.3. Factor structure: Auditors *Post* stage aspect ratings

Aspect	Component 1	Component 2
Cash flow	**0.722**	0.041
Liquidity	**0.808**	0.062
Capital structure	**0.753**	0.303
Profitability	**0.781**	0.134
Revenue	**0.530**	0.320
Financial flexibility	0.441	**0.592**
Industry Trends	0.425	**0.524**
Operational structure	0.213	**0.668**
Litigation	-0.167	**0.815**

In the six-aspect auditor negligence case (Experiment 2), there is less room for simplification. Even if the number of significant factors remained at two throughout the three decision-making stages, the total variance explained by the two factors increased from 58.9% in the *Pre* phase, to 65.4% in the *Leaning* phase, and 68.9% in the *Decision* phase. The structure of the two factors changes only slightly over the three phases. In the *Pre* and *Decision* phases, Component 1 is characterized by high factor loadings for *Safety Standards*, *Labor Negotiations*, and *Cash Problems of Biggest Customer*, whereas Component 2 is characterized by high factor loadings for *Toy Industry*, *Relationship with Suppliers*, and *Patent*. Thus, the six aspects are simply grouped into a bad-news dimension (C1) and a good-news dimension (C2). The only difference between the *Leaning* and the other phases is that *Patent* shifted into Component 1.

Thus, in the case of Experiment 1, we observe a gradual reduction in the number of significant factors as the decision makers progress through the decision stages; a gradual simplification of the knowledge structure as the decision makers make more sense of the decision problem and differentiate the alternatives. Again, a simpler mental model facilitates alternative differentiation and ultimately the decision.

18.3.4 Decision Criteria

In the introduction to this chapter, we suggested that decision makers may distort not only the underlying information, but also the decision criteria (thresholds) they use when making a decision. Davis and Ashton (2002) found that auditors tend to adjust their thresholds when the risks associated with negative decision outcomes are elevated. Phillips (2002) reports that participants in an auditing case distorted the *information evaluations* **during** the process of reaching a decision, and the *decision criterion ratings* **after** the decision. In the case of law, Thomas and Hogue (1976) found that jurors' decision criteria were distorted in response to the severity of the proposed penalty (*cf.* Kerr 1978).

Following Davis and Ashton (2002), we infer (*cf.* Lundberg and Nagle 2005) the decision thresholds from the participants' aspect ratings and their report choice. This novel procedure differs from direct elicitation of the thresholds from the participants. In contrast to Davis and Ashton who related report choice to the *auditors' reported probability that the client would continue as a going concern*, we relate report choice to the relative *amount of positive and negative information each participant associated with the particular 'client.'* Also, in contrast to Davis and Ashton, who computed a threshold for the whole group of auditors, we compute thresholds for the professional auditor subgroups based on years of experience and position in the firm (Staff, Manager, Partner). We also compute a threshold for a matching pool of 61 advanced auditing student participants. Again, we hypothesize that the point (based on the various aspects) where the auditor will be indifferent with respect to which report type to choose will be lower (indicating more tolerance for negative information) for more-experienced participants than for their less-experienced counterparts.

In order to simplify the model, we base the threshold computations on the average of the nine aspect ratings. This procedure is justified by the strong multicollineraity between the aspects (*cf.* Table 18.1), and results in a single measure of positivity/negativity. A model with this *average* as the sole independent variable and *report choice* as the dependent variable correctly classifies 92.1% of the report choices (97.9% for UNM and 75.8% for GCM, respectively) [Pseudo $\mathbf{R}^2$ = 0.7618 (Nagelkerke); χ^2-statistic on log likelihood ratio (N = 127) = 52.42 ($p < 0.001$)].

The model parameters from the four subgroups (three categories of professionals and students) are used to estimate the logit of the proportion of auditors who chose a going-concern modification ($\boldsymbol{p}$) as follows:

$$\text{logit}(\boldsymbol{p}) = \boldsymbol{\alpha} + \boldsymbol{\beta}'\boldsymbol{x}, \qquad (18.1)$$

where $\boldsymbol{\alpha}$ is the model constant, $\boldsymbol{\beta}'$ the vector of slope coefficients, and $\boldsymbol{x}$ the variable (the average). The proportion, $\boldsymbol{p}$, is calculated as

$$\boldsymbol{p} = [e^{(\boldsymbol{\alpha}+\boldsymbol{\beta}'\boldsymbol{x})}] \,/\, [1 + e^{(\boldsymbol{\alpha}+\boldsymbol{\beta}'\boldsymbol{x})}]. \qquad (18.2)$$

For example, the parameters for a model with all professional participants as one group were $\boldsymbol{\alpha}$ = 15.926 and $\boldsymbol{\beta}$ = - 0.299. Thus, an auditor whose aspect average is a quite positive 108.4 (on a scale from 0 to 125 - most negative to most positive-) is associated with a likelihood of choosing a going concern modification (labeled 1 in the model) of 0.000. Auditors whose averages are 56.1 and 43.9 are linked to much higher going concern likelihoods: 0.305 and 0.943, respectively. The overall threshold can then be derived, utilizing (18.2), by finding the $\boldsymbol{x}$ (the aspect rating average) for which the pool of auditors are indifferent with respect to report choice ($\boldsymbol{p}$ set to 0.5); where auditors are equally likely to choose an UNM and a GCM. For the auditor pool as a whole, this would result in a threshold of 53.35 (or 42.68% of the maximum positive score, and thus a point where the negative information outweighs the positive).

When this methodology is applied separately to each auditor group (based on years of experience and position in the firm) and to the student group, a rather clear picture emerges. The four thresholds associated with auditors distinguished by *years of experience* are depicted in Figure 18.4. The figure displays thresholds as a percent of the maximal aspect average score; 0.50 constituting a natural negative–positive cutoff. Interestingly, the thresholds are perfectly ordered by experience, with the lowest threshold associated with the most experienced auditors, followed in ascending order by those of auditors with five to ten years of experience, then by auditors with less than five years of experience, and finally by students lacking experience. In other words, the more experience an auditor has, the more willing s/he becomes to tolerate negative information before issuing a GCM, and consequently the more willing s/he becomes to take risks.

When the auditors are distinguished by position, the partners' and the students' thresholds again constitute the extremes. However, little difference is observed between the thresholds of managers and staff auditors. All in all, in accordance with Proposition 1, the point where the auditor is indifferent with respect to report choice indeed varies with the auditors' experience/position. It is, however, important to

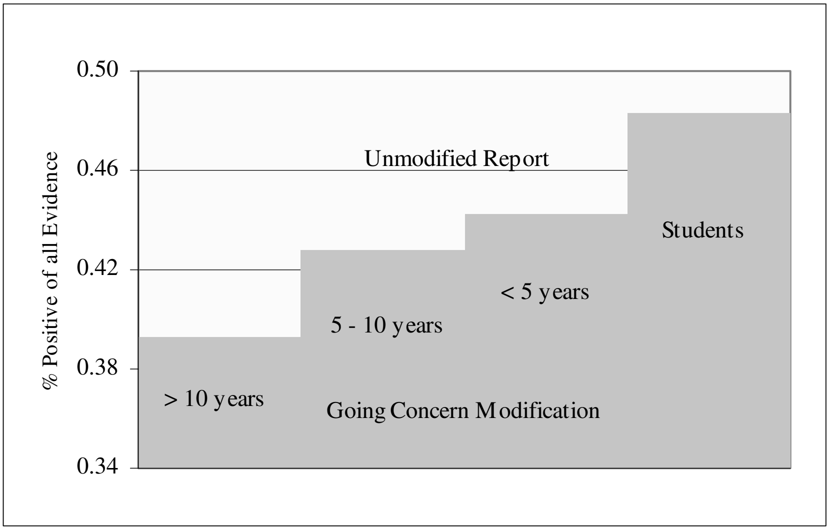


Figure 18.4. Going concern thresholds versus experience

note that increasing experience also makes the selection of a going concern modification less likely. As such, the relatively small group of partners is associated with a yet smaller number of negative reports.

18.4 Discussion

The complexities of inference-based decision making are made explicit through two multistage experiments set in complex and naturalistic decision environments. It has been argued that self-serving distortions potentially influence virtually all elements of decision making. This chapter spans components of the intricate process of making sense, and, in particular, decision makers' purposeful distortion of information during the process of reaching a decision and when justifying a decision. We also address issues related to the gradual simplification of the implicit task structure and experience-related differences in decision criteria.

Importantly, it is clear that the decision makers are unaware of their active information structuring. Distortion of information and criteria during the process of reaching a decision may result in biased or even faulty decisions, whereas distortion of information and criteria after the decision may influence similar decisions made in the future, and thus lead to faulty or impeded learning. Postdecision distortion makes decisions appear well justified, and, thus, may make decision makers feel unduly confident and may deter them from reflecting on their reasoning. In Phillips' (2002) terms, postdecisional distortion for a current decision becomes pre-decisional distortion for subsequent decision making. In a similar vein, Lundberg and Nagle (2002) show how spontaneous and feedback-induced attribute editing may cause a decision maker to view her/himself as more consistent and less revisionist than s/he actually is. However, in contrast to these detrimental facets, biases in information processing and threshold application also facilitate complex, inference-based decision making. The natural processes of information restructuring, for example, help make decision makers able to differentiate decision alternatives and ultimately make a decision. Similarly, biases associated with decision thresholds (Davis and Ashton 2002, Phillips 2002) may be both detrimental and facilitating. In other words, bias may be a necessary component in complex sense and decision making.

We forward the idea of using naturalistic experiments as a means for making complex, implicit processes explicit; and thus in an indirect way for supporting inference-based decision making. As most of these decisions happen in real time, occasional experiments run in the decision makers' respective organizations or during professional continuing education meetings, may constitute the most realistic support mechanism. They may, for example, alert the decision maker to the likely cognitive pitfalls at the various decision stages and to the consequences for the decision maker's learning from experience. Most notably, experience gained from experiments may, for example, identify if, and if so, how far one's preferences actually shifted in the course of making a decision (Simon *et al.* 2004). From the perspective of an individual decision maker and her/his organization, the present experiments show the importance of and inherent danger associated with initial favored alternatives, and how these early favorites are further supported as the decision maker restructures the underlying information. Something as simple as a decision log kept during the decision process can illuminate both the processes of early selection and the consecutive information restructuring. This log is likely to be a useful reference for a decision maker during the process itself, and for when s/he re-enters a similar decision domain in the future.

The present experiments also point to the biases that decision makers bring with them, most noticeably report base-rates. A decision maker and her/his organization may find it useful to benchmark individual choices to those made by peers. For example, the present experiments reveal that an auditor's experience and position in her/his firm influence her/his audit report choice. Importantly, the choices that people make, and the decision criteria they use, help shed light on the risks that they are willing to take.

In conclusion, we propose that organizations institutionalize recurrent experiments and/or simulations as light shed on the process of making inference-based decisions, in conjunction with practice and reflection, may be the best way to improve an individual's decision making, and to better understand the collective actions of the organization, and their associated risks.

Acknowledgement

The support of the US National Science Foundation (Grant # 0111969) is gratefully acknowledged.

References

Andersson P (2001) P1198: Software for tracing decision behavior in lending to small businesses. Behavior Research Methods, Instruments, and Computers 33:234-242

Andersson P (2004) Does experience matter in lending? A process-tracing study on experienced loan officers' and novices' decision behavior. Journal of Economic Psychology 25(4):471-492

Asare SK (1992) The auditor's going-concern decision: Interaction of task variables and the sequential processing of evidence. The Accounting Review 67(2):379-393

Bolger F, Wright G (1994) Assessing the quality of expert judgment. Decision Support Systems 21(1):1-24

Brézillon P, Pasquier L, Pomerol J-C (2002) Reasoning with contextual graphs. European Journal of Operational Research 136:290-298

Brownstein AL (2003) Biased predecision processing. Psychological Bulletin 129(4):545-568

Davis EB, Ashton RA (2002) Threshold adjustment in response to asymmetric loss functions: The case of auditors' "substantial doubt" thresholds. Organizational Behavior and Human Decision Processes 89(2):1082-1099

Holyoak KJ, Simon D (1999) Bidirectional reasoning in decision-making by constraint satisfaction. Journal of Experimental Psychology: General 128:3-31

Holyoak KJ, Thagard P (1997) The analogical mind. American Psychologist 52:35-44

Janis IL (1968) Stages in the decision-making process. In: Abelson RP *et al.* (eds.) Theories of cognitive consistency: A sourcebook. Rand McNally, Chicago, pp. 577-588

Kerr NL (1978) Severity of prescribed penalty and mock jurors' verdicts. Journal of Personality and Social Psychology 36:1431-1442

Kunda Z (1990) The case for motivated reasoning. Psychological Bulletin 108(3):480-498

Lowe DJ, Recker PMJ (1994) The effect of hindsight bias on jurors' evaluations of auditor decisions. Decision Sciences 25:401-426

Lundberg CG (2000) Made sense and remembered sense –Sensemaking through abduction. The Journal of Economic Psychology 21:691-709

Lundberg CG (2004) Modeling and predicting emerging inference-based decisions in complex and ambiguous legal settings. European Journal of Operational Research 153:417-432

Lundberg CG (2005) Models of emerging contexts in risky and complex decision settings. European Journal of Operational Research, (in press)

Lundberg CG, Nagle BM (2002) Post-decision inference editing of supportive and counterindicative signals among external auditors in a going concern judgment. European Journal of Operational Research 136:264-281

Lundberg CG, Nagle BM (2005) The effect of experience on the identification of and tolerance for negative evidence in a going concern situation, (manuscript under review)

Montgomery H (1983) Decision rules and the search for a dominance structure: Towards a process model of decision-making. In: Humphreys PC, Svenson O, Vari A (eds) Analyzing and aiding decision processes. North Holland, Amsterdam, pp. 343-369

Payne JW, Bettman JR, Johnson EJ (1993) The adaptive decision maker. Cambridge University Press, New York

Phillips F (2002) The distortion of criteria after decision-making. Organizational Behavior and Human Decision Processes 88(2):769-784

Sá WC, West RF, Stanowich KE (1999) The domain specificity and generality of belief bias: Searching for a generalizable critical thinking skill. Journal of Educational Psychology 91(3):497-510

Shanteau J, Stewart TR (1992) Why study expert decision-making? Some historical perspectives and comments. Organizational Behavior and Human Decision Processes 53(2):95-106

Shultz TR, Katz JA, Lepper MR (2001) Clinging to beliefs: A constraint-satisfaction model, Poster, 23rd Annual Conference of the Cognitive Science Society, Edinburgh, Scotland. http://www.hcrc.ed.ac.uk /cogsci2001/poster-2.html

Simon D, Krawczyk DC, Holyoak KJ (2004) Construction of preferences by constraint satisfaction. Psychological Science 15(5):331-336

Simon HA (1960) The new science of management decision, Harper & Row, New York

Slovic P (1995) The construction of preference. American Psychologist 50:364-371

Slovic P, Finucane M, Peters E, MacGregor DG (2002) The affect heuristic. In: Gilovich T, Griffin D, Kahneman D (eds) Heuristics and biases: The psychology of intuitive judgment. Cambridge University Press, New York, pp. 397-420

Sniezek JA, Schrah GE, Dalal RS (2004) Improving judgement with prepaid expert advice. Journal of Behavioral Decision-making 17(3):173-190

Svenson O (1992) Differentiation and consolidation theory of human decision-making: A frame of reference for the study of pre- and post-decision processes. Acta Psychologica 80:143-168

Svenson O (1998) Multi-criteria decision aids and human decision-making: Two worlds? Letter to the editor. Journal of Multi-Criteria Decision Analysis 7:352-354

Svenson O (1999) Differentiation and consolidation theory: Decision-making processes before and after choice. In: Juslin P, Montgomery H (eds.) Judgment and decision-making –Neo-Brunswikian and process-tracing approaches. Lawrence Erlbaum Associates, London, pp. 175-198

Thagard P (2001) Coherence in thought and action. The MIT Press, Cambridge, MA

Thomas EAC, Hogue A (1976) Apparent weight of evidence, decision criteria, and confidence ratings in juror decision-making. Psychological Review 83:442-465

19

A Role for Information Portals as Intelligent Decision Support Systems: Breast Cancer Knowledge Online Experience

Frada Burstein, Sue Mckemmish, Julie Fisher, Rosetta Manaszewicz, and Pooja Malhotra

Enterprise Information Research Group, Monash University, Melbourne, Australia ,
PO Box 197, Caulfield East, 3145, Victoria, Australia, Telephone: +61 9903 2208
e-mails: {Frada.Burstein; Julie.Fisher; Sue.McKemmish}@sims.monash.edu.au

In this chapter we review the knowledge-based view on decision support and argue the emergence of a new type of intelligent decision support system – an intelligent gateway for supporting specific knowledge needs. The modern view on decision support and expert systems has shifted from considering these as purely analytical tools for assessing best-decision options to seeing them as a more comprehensive environment for supporting efficient information processing based on a good understanding of the problem context. Such intelligent decision support systems incorporate problem-domain knowledge to improve their information processing and provision capabilities. More recently, information portals have been proposed as tools for matching users' information needs in order to enhance their decision-making ability. This chapter looks at portals as new types of intelligent decision support systems, which use problem-domain knowledge in order to improve efficiency in information provision. The main focus of the chapter is on suggesting mechanisms for implementing intelligent decision support capabilities in a healthcare portal, which seeks to deliver personalized information to support efficient decision making. BCKOnline, a healthcare portal built around breast cancer information, is described as an example of such implementation.

19.1 Introduction

Intelligent decision support systems were introduced as an enhancement of traditional decision support systems, when access to the domain knowledge is necessary for better informed decision making. From a technological aspect they are viewed as a class of decision support system that includes a knowledge base as an encoded component of decision support architecture (Power 2002). Expert systems and other Artificial Intelligence (AI)-based technology were commonly utilized as mechanisms for storage and provision of domain knowledge (Turban and Aronson 1995, Klein and Methlie 1995).

The modern view on decision support and expert systems has broadened from considering these as purely analytical tools for assessing best-decision options to seeing them as a more comprehensive environment for supporting efficient information processing based on good understandings of the problem context. As a result of this intelligent decision support systems (IDSS) are expected to play the role of a mediator between the user – *e. g.* the decision maker - and the task they perform (Burstein *et al.* 1994). Implementation of such decision support processes assumes the existence of an explicit model of the decision situation (task) and not only a domain knowledge base, but some mechanism for dynamically adapting it to user needs in specific decision situations (Linger and Burstein 1997). The core focus of IDSS functionality has thus shifted from capturing expert knowledge to continuous knowledge management and knowledge work support (Burstein and Linger 2003). There is still a provision for access to the domain knowledge-base, however, the user has much more flexibility and authority to express her specific information requirements and expect a personalized response. Such an approach assumes that an intelligent decision support system "*… should behave like human consultants, supporting decision makers in understanding, expressing and structuring their problems*" (Angehrn and Luthi 1990).

Information provision for decision support over the Internet presents new opportunities and challenges for representing domain knowledge to assist users in retrieving personally relevant information. The fact that vast information resources are now published and available online has made a significant impact on the way people seek information to support their decision-making. In particular, reliance on Internet health information is widespread and increasingly encouraged by Governments and other health agencies (Health Online 2004). This reliance on online information is creating concern about the quality and authoritativeness of this information (Christensen and Griffiths 2000, Schmidt and Ernst 2004). Given that access to information is a fundamental component of decision making the challenge is to provide information that is relevant, timely, accurate and as far as possible meets the dynamic information needs of the user. In such an environment intelligent decision support systems can play a role in communicating the right knowledge to the right user.

More recently, information portals were proposed as tools to enhance users' decision-making ability. Holsapple and Whinston (1996) describe a knowledge-based approach to decision support. They suggest decision support is a process of manufacturing new knowledge by enhancing users' ability to update their prior knowledge based on the information provided by the system. The ability of the user to appreciate and internalize information they require for making decisions depends on the context and individual circumstances they face when making a decision. Information portals have a potential for facilitating such knowledge-based approaches as tools that utilize users' information needs model (profiles) to enhance their decision-making ability (Firestone 2003). Information portals connect people with information and applications they need for performing tasks. Unlike a conventional website, a portal should support both push (subscription) and pull (search) functions in assisting users to gain access to essential situated information. In this sense portals can play the role of a new type of intelligent decision support system, which use problem-domain knowledge in order to provide differentiated information access.

Moreover, in addition to internal data and knowledge bases, portal architecture allows access and provision of information from external sources, which are deemed useful for supporting decision making. In essence, a portal interface serves as a gateway to necessary information resources. Such broadening of the sources of information leads to classical information management problems related to rigor, speed and completeness of selection, appraisal and interpretation of resources made available through the gateway. The challenge of such an approach is in taking a user-centric rather than an information-centric focus in these problems.

In this chapter we review the knowledge-based approach to decision support and argue the emergence of a new type of intelligent decision support system – an intelligent gateway for supporting specific knowledge needs -. This chapter looks at portals as new types of intelligent decision support systems that incorporate problem-domain knowledge in order to improve efficiency in information provision. The main focus of the chapter is in suggesting how certain characteristics of an intelligent decision support system can be implemented in a portal, which seeks to deliver personalized information to support efficient healthcare decision making. We illustrate how the Breast Cancer Knowledge Online (BCKOnline) portal addresses the challenge of meeting the diverse information needs of women with breast cancer and their families through the provision of timely, relevant information to support decision making via a portal (Burstein *et al.* 2005). The role that has been envisaged for the BCKO portal is one of an online resource serving multiple purposes to support informed decision making, provide authoritative, relevant health care and related information to users, to manage the quantity of information presented and to provide information about the quality and provenance of the information accessed using metadata.

We propose metadata-driven mechanisms for user-centric resource description, which can be utilized to better meet the information needs of users. Such a description requires expert knowledge within the domain and a good understanding of the context the resources will be used in, including the changing and diverse needs of users. The intelligence of such a system depends in part on how well it knows what the users' expectations are and on the quality of the information it provides access to. In the BCKOnline portal, expert knowledge about user needs, relevant resources and their quality, and the outcomes of user information needs analyses are captured in descriptive metadata, which can be defined as standardized information about the content and context of information resources, including information about the users. Metadata schemas define metadata elements that can be used to describe information resources for purposes such as resource discovery and delivery, and provide guidelines for the creation of metadata records that essentially provide a catalog of information resources available online.

19.2 Decision Support Systems and Expert Systems

Decision support systems (DSS) over time have become increasingly sophisticated, making use of models from a variety of disciplines ranging from artificial intelligence, operations research, and management science. Systems that use artificial intelligence techniques are often referred to as expert systems (ES) or knowledge-based systems (Dhar and Stein 1997). This section reviews decision support and expert systems concepts in relation to the potential of an information portal to play these roles.

A DSS is an interactive, computer-based information system that utilizes decision rules and models, coupled with a comprehensive database (Turban and Watkins 1986, Sprague and Carlson 1982). A properly designed decision support system is expected to improve the effectiveness of a decision maker by providing a powerful modeling and data analysis tool.

An ES on the other hand, is a computer program that includes a knowledge base containing an expert's knowledge for a particular problem domain, and a reasoning mechanism for propagating inferences over the knowledge base (Turban and Aronson 1995). Updating and maintaining the knowledge base and enhancing the capability of the inference engine are central to an ES (Raggad and Gargano 1999). Knowledge-based systems are a broader term to define computerized systems, that capture some expertise in a computable form and make it available to the user needing expert advice about this context-specific knowledge. This knowledge can be sourced from individual experts, the collective explicit knowledge of the professional group or some other published material.

There are several fundamental differences between these two technologies. For example, DSS database contains facts whereas in ES the knowledge base contains, in addition to facts, procedures for problem solving. An ES by definition exhibits reasoning capability, DSS does not. Furthermore, in the DSS environment the user asks the system questions before reaching a decision, whereas in the ES environment the system asks the user questions before reaching a decision (Bidgoli 1993).

If a portal is to play a role in addressing user needs effectively, it requires an integration of decision support system features and some aspects of expert-system functionality. Such integration has been addressed by introducing the concept of intelligent decision support system (Turban and Watkins 1986).

19.3 An Intelligent Decision Support System

Despite some definitional differences there is a strong sense in the literature that AI-related technologies such as expert systems can improve the quality of today's DSS and *vice versa*, for instance existing expert systems can be used as independent computerized systems similar to DSS systems, advising users on a specific problem area. The integration of ES and DSS can offer a more balanced system containing expert knowledge, as well as reasoning and explanation capabilities, with greater emphasis on the end-user profiles. DSS/ES integration benefits can be realized along several dimensions: the ES contribution, DSS contribution, and the synergies resulting from the DSS/ES contribution. Turban and Watkins (1986) describe possible theoretical models of integration by adapting an existing DSS system to perform in an ES style. Such adapted systems are considered by many to be *intelligent decision support systems* (IDSS) (Turban and Watkins 1986, Hollnagel

1986, Norman 1986, Bidgoli, 1993) with the focus on the functioning of 'man and machine' together. Despite the complexity of the integration process, recent literature suggests there are promising signs for the integration of DSS and expert systems.

An IDSS is more of a cognitive rather than a technological system. The fundamental difference is that even basic characteristics of intelligence cannot be captured in mechanistic terms (Hollnagel 1986, Burstein *et al.* 1994, Linger and Burstein 1997). Conversely, a cognitive system,

> *produces 'intelligent action', that is, its behavior is goal oriented, based on symbol manipulation and uses knowledge of the world (heuristic knowledge) for guidance. Furthermore, a cognitive system is adaptive and able to view a problem in more than one way. A cognitive system operated using knowledge about itself and the environment, in the sense that it is able to plan and modify its actions on the basis of that knowledge. It is thus not only data driven, but also concept driven. (Hollnagel and Woods, 1983)*

From this definition, a number of characteristics of cognitive systems may be derived. Of interest to this chapter are the minimum defining characteristics required of an IDSS as a cognitive system and an expert/decision support system that can then be implemented in a portal. In this context, a system requires functionality that is currently absent from the traditional model-based decision support systems (DSS); mainly learning, memory and reasoning. Portal-based implementation provides an opportunity to collect and store context-specific knowledge with the aim of building in learning capabilities in the future based on analyzing patterns of use of the existing and desired content of the facility.

19.4 Information Portals

An information portal is not a new concept. There are numerous portals acting as 'gateways' to information and services as well as a taxonomy of the various types of portals based on their functionality (Firestone 2003). Enterprise Information Portal, EIP, is the most appropriate type for our purpose. An EIP can be defined as "… *a single gateway connected by a server that connects people with information. It allows access to services, software applications and a variety of information*" (Harvard Computing Group 2002). Shilakes and Tylman (1998, p. 1) define information portals as "*applications that unlock internally and externally stored information, and provide users with a single gateway to personalized information to make informed decisions*". In this sense an EIP can be seen as a decision support tool, which can take an active role in helping the users define and meet their information needs. To achieve this goal a portal requires a special interface and built-in knowledge about the problem domain in order to guide the users intelligently through the process of locating and delivering quality information sources. A sceptic would say that the term 'portal' no longer provides any definite meaning given the number of software products that are now touted as having some type of portal functionality. However, there is a role for portals as intelligent decision support systems that has not been fully explored yet.

The emphasis in this chapter is on a portal as a decision support facility and a new type of expert system that is capable of providing comprehensive support to meet the diverse and changing needs of individual users and to 'add value' by enabling users to judge the quality and reliability of the information provided. Such a portal will not make decisions for the users but will provide them with the kind of information they need to make informed decisions for themselves.

19.5 IDSS Functionality for Portal

This section discusses a minimum functionality that a portal should have to make it an IDSS. The following major characteristics essential for a portal as an IDSS:

- knowledge repository and memory,
- intelligent information retrieval;
- information classification and prioritizing;
- adaptivity and personalization;
- explanation facility;
- metadata broker.

A portal with these features is capable of providing users with personalized information to make informed decisions, as per the general definition of a portal presented earlier in this chapter.

19.5.1 Knowledge Repository and Memory

An information portal plays the role of a gateway to relevant information in the form of a virtual collection of information resources, selected with reference to expert domain knowledge and user needs. To satisfy this role it has to have some knowledge of its purpose, the context it is built to serve, its audience of users and the information resources relevant to them. As such a portal may not contain information resources within its physical infrastructure, but it has to have a knowledge base, which contains the context model, user model, some mechanism of matching between the two, and links to the virtual distributed collection of information resources. This knowledge repository can initially be created based on problem domain knowledge representation. However, due to the dynamic nature of knowledge this repository needs to be constantly updated and maintained. Such maintenance is an essential feature of a true intelligent DSS. It converts a knowledge repository into a dynamic memory system (Linger and Burstein 1997).

19.5.2 Intelligent Information Retrieval

The goal of information retrieval (IR) is to retrieve only the documents relevant to a user's information needs. A better way to understand the characteristics of information retrieval is by studying it in relation to data retrieval. Information retrieval is quite often mistaken for data retrieval. Although the boundary between data retrieval and information retrieval is often quite vague, nevertheless, different ranges of complexity are associated with each mode of retrieval (Rijsbergen, 1979). In document retrieval (DR) one normally looks for an exact match, checking whether an item is or is not present in the file. In information retrieval, generally one wants to find those items that partially match the request and then select from those a few of the best matching ones. As a result in DR the query is generally a complete specification of what is wanted, in IR it is invariably incomplete. The extent of the match in IR is assumed to indicate the likelihood of the relevance of that item. One simple consequence of this difference is that DR is more sensitive to error in that an error in matching will not retrieve the wanted item, which implies a total failure of the system. In IR, small errors do not significantly affect system performance.

Consequently a good IR model is one that gives each document a relevance estimation as close as possible to the user's own relevance judgment (Nie and Lepage, 1998). This idea of *relevance* is what is at the center of information retrieval. The purpose of a retrieval strategy is to retrieve all the relevant documents and at the same time retrieving as few of the nonrelevant ones as possible (Rijsbergen 1979).

Furthermore, the cognitive relevance for information retrieval is not just based on the relationship between a document and the topic of a query but rather it is a 'document-query' relationship within a certain context and as such establishes *situation-dependent relevance* (Nie and Lepage 1998). A cognitive IR model can be contrasted with a computational model as shown in Figure 19.1.

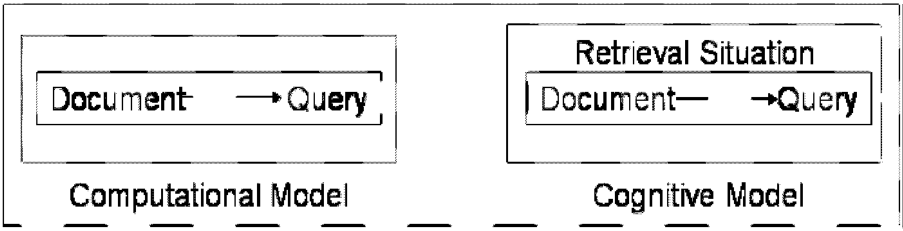


Figure 19.1. A relationship between computational and cognitive model of information retrieval (adapted from Nie and Lepage 1998, p.18)

In principle, although situation-dependent retrieval seems simple and logical, it is very difficult to implement. Quite often it relies on classification-based technology that has evolved as part of knowledge representation research and can be based on some knowledge representation formalism, for example, semantic networks (Gregor 1991).

The success of the implementation of a true context-based retrieval system depends on the quality of the cognitive model and ability to explicitly match the user's context with the captured model in the context of the document. Some of this context-related information, including expert-domain knowledge about information resources and user needs, can be captured in metadata. For example, the AGLS metadata schema (AGLS 1998) includes an element called Audience, which defines who this information may be relevant to. Many portal search engines combine a full text search with a metadata search facility. So if information about the context is expressed as part of a metadata record, it can help to increase search efficiency and retrieval as well as reduce search time. Context-related metadata can also be useful when displaying to the user a summary of search results rather than the entire document.

19.5.3 Adaptivity and Personalization

An adaptive system is continually attempting to configure itself so as to match the input data. It is a flexible yet rigid system. That is, although it is always adapting to mirror the arriving data, it does so by means of precoded existing knowledge and within existing configurations. A special consequence of adaptivity is multiviability (Hollnagel 1986). This means that a system can reach its goal in more than one ways, and it implies that the choice of a particular way is based on knowledge of the characteristics and requirements of the current situation, rather than being random. The ability to choose an appropriate way to the goal means that the system is intelligent (Hollnagel 1986) agreeing with the general notion that intelligence characterizes the means rather than the end. A fundamental and essential aspect of adaptive systems behavior is learning (Norman 1986). Raggad and Gargano (1999) defined learning as the ability of the ES to improve system recommendations based on experience. They also provide a model for ES learning that incorporates three components: a learner, an evaluator, and a testing case store as illustrated in Figure 19.2.

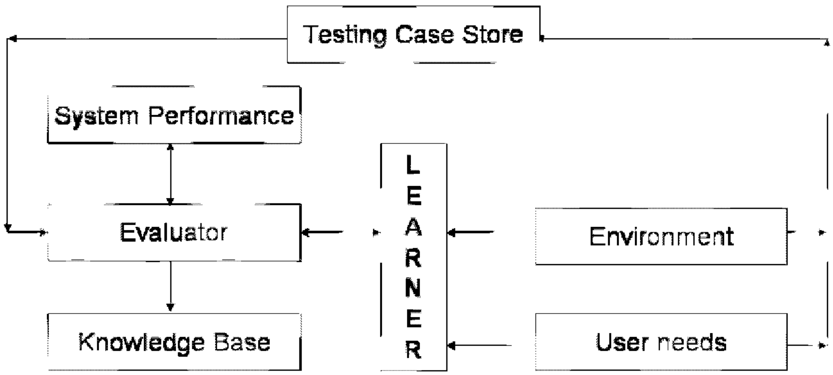


Figure 19.2. Expert system learning model (Raggad and Gargano, 1999)

The main functionality outlined in their model is that information is fed to the learner, which is then evaluated by the evaluator by comparing it with a number of random cases taken from the testing case store. If the performance of the ES is high then the new knowledge is added to the knowledge base.

Personalization has the ability to provide differentiated information access to users and is one of the major features, that distinguish information portals from conventional websites. In this sense, the model presented in Figure 19.2 can be applied to describe the operational behavior of a portal. A Testing Case Store is equivalent to predefined user profiles, which are used by a portal interface to identify relevant information from the knowledge base. Existing intelligent portals "learn" about the information needs of individual users by analyzing patterns of previous information seeking behavior. Thus commercial services such as amazon.com can provide individual users with information about new publications that might be relevant to them on the basis of their previous information seeking and purchasing behavior. Portals can also learn about the user needs and relevant information resources by referring to user-centric resource descriptions (stored in its metadata repository as outlined below) and predefined user profiling, based on qualitative user information needs analysis and expert-domain knowledge, which can also be incorporated in its knowledge base.

19.4.4 Information Classification and Prioritizing

Users can become overloaded and distracted when many events occur in a short time period. This can happen when they need to either continually change focus to deal with new arriving information or focus upon what might turn out to be not critically relevant aspects of the situation (Norman 1986). This is where an IDSS can play a major role. The goal of an IDSS should be to classify and prioritize the information and advise the user on various aspects of the quality of the information. Classification can be based on a range of techniques including the implementation of metadata/classification schemes based on user information needs analysis and expert domain knowledge, methods associated with semantic networks, and classification based on text mining and ES approaches.

A number of Web-specific studies highlight various aspects of human behavior focusing specifically on Web-based information seeking (Ford *et al.* 2002). The study of users' interaction with Web search engines is an important and emerging research area with implications for the development of more effective Web-based human-computer interaction models, search engines and interfaces (Spink *et al.* 2000). Ranking is particularly important in web-based information seeking insofar as web searchers on average, display only the first ten retrieved items (Ford *et al.* 2002) where a page is a group of ten results. Ranked systems are best used for higher recall than precision (Jones 1999). The advantage of relevance ranking is that although it does not guarantee complete recall, it brings more relevant documents to the top of the list. The larger and more heterogeneous 'the collection' of documents the more difficult it may be to achieve sufficiently high precision to ensure that relevant retrieved items will be presented within a number of pages acceptable to many searchers. Thus, arguably search techniques like Boolean searching that lack a ranking capability are at a disadvantage in the context of Web searching on portals. Jansen and Pooch (2001) in reviewing these studies argue:

> *.....the vast majority of web searchers use approximately two terms in a query, have two queries per session, do not use complex query syntax, and typically view no more than 10 documents from the results list. Use of Boolean operators in web queries is almost non-existent, ranging from 2% to 8%. (Jansen and Pooch 2001)*

Classification and prioritizing information is used in portals as a part of a mechanism for differentiated information provision. The amount and form of the information to be presented to the user can also be controlled through this mechanism and initial dialog with the users at the time they define their profiles.

19.4.5 Explanation Facility

In principle an intelligent system needs to be capable of explaining to users both the knowledge they contain and the reasoning processes they go through. Since the advent of advice-giving intelligent computer systems, explanation facilities have been one of their important and valued features (Gregor and Benbasat 1999). Explanations, by making the performance of a system transparent to its users, influence user acceptance of intelligent systems and improve users' trust in the advice provided.

In a portal context users may be interested to know why the information provided is regarded as relevant and valid. Some form of explanation relating to the criteria for resource selection and basis of resource description is therefore necessary. It can incorporate ratings based on assessments from the previous users of the information. This facility can also be implemented as part of the quality-assessment procedures captured in a subset of the metadata elements. In addition to providing information about the quality, reliability and authoritativeness of the resources accessible via the portal, there also needs to be an explanation of how quality is defined. Sufficient information about the reliability, quality and authoritativeness of the portal itself, its intelligent features and the value-added information it contains also needs to be provided to enable users of the portal to assess its usefulness to them in terms such as fitness for purpose.

19.6 Metadata as a Mechanism for Intelligent Decision Support

The role of metadata in DSS implementation has been very limited. Power (2002), while identifying metadata as an essential component of Data-Driven DSS, suggests it as a mechanism to capture mainly semantic information about operational data. He limits its content to factual information about the nature and origins of the data, as well as other elements standard to the requirements of a data dictionary, *e. g.* maintenance information and data formats. In making a distinction between data-driven, document-driven and knowledge-driven DSS, Power recognized the role of metadata in assisting data and document retrieval, but did not recognize the opportunity to use metadata in the more extensive, descriptive way discussed in this chapter in order to support users' resource discovery and decision support requirements. We argue that in an information portal with IDSS functionality, metadata can provide a mechanism for user-centric resource description, which can be utilized to better meet the users' information needs.

In Web environments, information discovery and delivery is increasingly being facilitated using metadata descriptions of information resources as a mechanism for describing context, content, structural and management information about the resource (Asprey and Middleton 2003). Metadata can also be used to describe the target audience for selected information resources and to develop profiles of users and their information needs. It can also capture patterns of use. Such metadata is partly already available in the context of Internet resource description due to Internet initiatives such as the Dublin Core Metadata Initiative (DCMI, 1995-2003) and the Dublin Core-based Australian Standard AGLS metadata schema.

One of the barriers to extending the use of metadata is that its creation is resource intensive. Some metadata elements can be readily generated through automated means, the creation of metadata that assists in establishing the relevance of a resource, however, its target audience or its fitness for purpose needs to draw on expert-domain knowledge and usually entails direct human involvement. A portal in IDSS role needs to incorporate the functionality of a metadata broker, which uses some of the intelligent features to create or assist the creation of high-quality metadata. The kind of metadata broker envisaged is drawn from the work of the Clever Recordkeeping Metadata Project (CRKM, 2003-5). The components of such a metadata broker include a metadata schema, a standardized set of metadata elements; a metadata repository comprising a catalog of metadata records describing information resources; and metadata translation and transformation services that use tools for automatic capture, repurposing or generation of metadata and updating of resource descriptions. The incorporation of expert-system approaches, text mining and supervised learning procedures would greatly enhance the power of such tools.

In the next section we describe an illustrative example of the healthcare information portal, BCKOnline, which has been developed based on user-centric metadata resource description.

19.7 IDSS Approach for BCKOnline – A Case Study

The BCKOnline portal aims to optimize efficiency in providing users with relevant information. This is achieved by implementing a metadata repository and user-centric information resource description. The BCKOnline portal as an interactive Web-based personalized information system exhibits some basic features of an IDSS as described above. This section analyses to what extent the BCKOnline portal supports the functionality of an IDSS.

19.7.1 Metadata Repository of User-centric Resource Descriptions

To be user sensitive, the portal needs to know the users's differentiated breast cancer information needs, and it needs rich contextual information about breast cancer information resources. Most online search engines rely on free-text searching - they examine and index the content of online documents, looking for word patterns and associations -. Given the highly contingent needs of women with breast cancer, and their concerns with the quality and reliability of the information they are seeking, we have developed complex user profiles based on extensive analysis of the diverse characteristics and differentiated information needs of members of the breast cancer community. These profiles have been used to develop a set of descriptors; a metadata schema, which will enable breast cancer information resources to be cataloged in an online database with reference to the information needs of their target audiences.

Usually online catalogs or metadata repositories contain descriptions of information resources that are resource-centric - describing resources in terms of attributes like Author, Title, and Subject, but not in terms of the needs of the target audience -. The BCKOnline portal contains a metadata repository, which is a catalogue that describes resources in this conventional way, but also includes 'user-centric' resource descriptors. For example, the metadata stored includes information about the target audience, how to access the resource, restrictions associated with its use, and information about its quality. By referring to these descriptions of information resources in its metadata repository, the portal will identify and retrieve resources with greater precision and hence relevance to the individual user. For example, treatments vary according to disease stage, hence the woman faced with advanced disease will require different information to a woman with early breast cancer. Younger women are more likely to be concerned with issues of fertility and self-esteem as a result of chemotherapy treatment and/or surgery (Thewes *et al.* 2003).

The nineteen element AGLS Metadata Schema (Australian National Standard) is based on and extends the Dublin Core set of metadata elements, an international code that was designed to facilitate resource discovery on the Internet. The schema used in the BCKOnline metadata repository adopted fifteen of the AGLS metadata elements, and added a new Quality element and qualifiers with a related encoding scheme. Additional qualifiers were defined for the Audience element, and sector specific encoding schemes were added for the Audience, Type and Subject elements. Table 19.1 presents the metadata elements used to describe resources in the BCKOnline repository.

The new Audience qualifiers and the Audience Encoding Scheme were defined with reference to user profiling based on the needs of the target audience as identified in the user information needs analysis and the current literature (Williamson and Manaszewicz 2003). The portal enables the user to build a profile based on the attributes defined in the Audience element qualifiers, and then, by searching on the Audience element in the metadata repository, matches the selected profile with resources that contain information relevant to the target audience represented by the profile. User profiles can be built from the following: age (Under 40: 40-49: 50 –69: Over 70); disease stage (early breast cancer: recurrent breast cancer: advanced breast cancer); information preference (plain/brief: plain/detailed: scientific/brief: scientific/detailed) and user type (self: child: friend: partner/spouse: parent). For example, if a user selects 'plain' and 'brief' as a presentation format, then her search will not retrieve medical journal articles. If a young woman is looking for personal stories and accounts the resources retrieved will be of stories from women in her age group.

Table 19.1. A summary of the BCKOnline metadata schema

Elements	Qualifiers/Encoding Scheme	Elements	Qualifiers/Encoding Scheme (ES)
Creator		Format	
Publisher		Date	
Contributor		Identifier	
Availability		Rights	
Title		Source	
Language		Relation	
Subject	BreastCare Victoria Glossary BCKOnline Disease Trajectory BCKOnline Key Words ES	Type	Medical; Supportive; Personal/ BCKOnline Category ES
Audience	Age group Disease stage User type Locality Information preference BCKOnline audience ES	BCKOnline Quality ES	Credentials Review process Evidence-based Purpose Balance Currency References Narrative/ BCKOnline ES

The Quality element and related Encoding Scheme were added to deal with the breast cancer community's requirements relating to reliability and quality, while its qualifiers were developed using a conceptual analysis framework derived from examination of existing quality standards, *e. g.* Health on the Net (2004), AMA Guidelines for Medical and Health Information Sites (2000), HiEthics (2004), Eysenbach and Kohler (2002), McKemmish *et al.* (1999), in addition to analysis of user information needs relating to quality. This element enables the portal to provide the user with a narrative report highlighting the essential characteristics of the 'quality' of the individual item – its provenance, authoritativeness and currency.

The AGLS Schema references a number of encoding schemes to assist users of the Schema in assigning metadata values to elements. For medical subject matter, AGLS specifies use of MESH, an international standard for the classification of medical knowledge. As its applicability is limited for consumer use in relation to some specific diseases, the BCKOnline Metadata Schema includes two additional sets of indexing terms specifically geared to breast cancer diagnosis, treatment and management, and the terminology used in Australia. The BCKOnline Disease Trajectory Encoding Scheme also provides the basis for one of the three search strategies available to users of the portal – the capacity to search on terms relating to detailed phases in the disease trajectory of breast cancer. Finally, the BCKOnline Metadata Schema adapted the Type element to enable users of the portal to narrow their search to include only resources of a medical, supportive or personal nature (Burstein *et al.* 2005).

19.7.2 BCKOnline Implementation

The prototype portal has been developed using the tool called HotMeta, which focuses on the use of metadata to increase the level of precision and recall for Internet search engines (Renato and Waugh 1997). HotMeta (HotMeta 2004) works through a Web interface. It integrates with MetaEdit, the metadata editor component to produce metadata or cataloguing records of the selected resources. HotMeta provides interfaces to enable users to browse, find and search the metadata records stored in the repository. It supports Dublin Core and AGLS metadata but also allows for customized metadata schema. The Metadata Search Engine dynamically generates Webpages of the search result. Figure 19.3 presents a screen shot of the profile page with the different options for searching.

Users may select by clicking the profile that best describes who they are, then users select the type of information they want and then they enter the search term. Users have 3 search options: using the profiles, using breast cancer topics defined with reference to a detailed breakdown of the stages in the disease trajectory (based on the BCKOnline Disease Trajectory Encoding Scheme) or a simple search without any profiles. Users may also elect to select only some of the profiles for example, age and disease stage.

Figure 19.3. Screen shot of the portal profile page

Figure 19.4. BCKOnline search result output

The search results screen presents the user with a list of records together with the search parameters, *e. g.* profile elements, search terms, information type and numbers of records matching the search. Each record is described by the title, short abstract, type of resource (*e. g.* medical, supportive, personal), and a narrative quality report. (see Figure 19.4 for example output screen).

If the user wishes to view the whole metadata description of the resource it is available through the follow-on hyperlink situated below each record.

19.7.3 Evaluation of the BCKOnline Portal

To assess the effectiveness of the portal from the user's perspective usability evaluations and focus groups were undertaken. "The primary goal of a usability test is to improve the usability of the product that is being tested".(Dumas and Redish 1994, p 22). Dumas and Redish (1994) argue that those who participate in usability evaluations should be people who are going to use the product. Sixteen women participated in two usability evaluations of the portal after the first test changes to the portal were made. A second usability evaluation was then conducted. A low number of participants is acceptable for a test such as this as research suggests that between five and eight users will generate useful results (Nielsen and Molich 1990, p.156).

The purpose of the usability evaluation was to assess both the functionality of the portal and to determine its effectiveness from the perspective of women with breast cancer. Initially participants were to spend 30 to 40 minutes exploring the portal however, it became evident that the women were so engrossed in using the portal that this phases lasted for an hour and a half. The usability questionnaire contained questions often requiring a free text response as well as scaled responses. Nine of the women participating in the second evaluation also participated in two focus groups (Hackos and Redish 1998). The purpose of the focus groups was to explore issues with the women, which could not be covered adequately with a questionnaire. For example, how interested they were in the quality of the information provided and whether the information they accessed had helped in any decisions they had to make.

The results of the evaluation are presented in the context of the key issues the women with breast cancer raised in the initial focus groups described earlier. The participants were from all the age groups with the majority (7) in the age group 46-55. Most participants had some experience with the Internet (7) or were very experienced (3). Eleven had breast cancer and seven indicated they were in the early stages.

Overall usability: All the women found the site easy to use and most (7) rated their enjoyment using the site as high (scale of 1 'Did not enjoy using the site' to 5 'High level of enjoyment'). When asked what would make them return to the site, the following responses were typical: 'Finding more information on particular topic', 'wanting to get new info, rechecking info, and getting info for other women with B.C.', 'Ease of search, links available, note of quality of information', 'It is easy to navigate… simple enough for women with the most basic computer knowledge'. When asked the extent to which they agreed or disagreed with the statement 'I would recommend the system to other women with breast cancer', 11 of the women agreed strongly with the statement. These responses have been further confirmed through many other informal comments we have received through e-mail.

Avalanche of information: Most women in the original focus groups highlighted the problem of retrieving too much information from Internet searches. With 500 carefully selected and tagged items the portal provides enough variety and coverage of topics to satisfy the users information needs without swamping them as this comment reflects Great work! Generally an excellent tool to help people navigate the abundance of information available on the Web. Another said it was good not to have too many choices .

Information relevance: Information relevance was another important issue raised by the women. The BCKOnline portal addresses this through the careful metadata tagging of individual resources to specific profiles as described earlier. Although the number of information resources is relatively low, providing users with the ability to select resources appropriate to their perceived needs via the 'personalized search' interface and the 'information preference' option has resulted in highly relevant resources being retrieved. It was clear from the comments that volumes of information aren't necessary if users retrieve information that meets their requirements, as illustrated by this woman, *for me it was very relevant. I got to straight where I needed to go.*

A major objective of this project was to develop a system that would allow users to describe themselves through profiles (*e.g.* age, disease stage) and their information preferences (*e.g.* brief, scientific language) so the resources selected were more likely to meet their needs. All the women participating in the usability evaluation except one, elected to search using the profiles and 7 of the women indicated this was the most valuable search mechanism. The comments from the women highlight the importance of providing relevant tailored information.

At different stages you need different things and that's what I like about that icon page [profile page].

Participants varied in the depth and/or brevity of the information they preferred. For some, superficial and generalized information about a particular drug was not helpful. What they required was the in-depth content provided via medical journal articles. The interface facilitated this as is evidenced by the following comment: *I want the most detailed medical information. And that takes you to the level of information the depth of information that you're after. That's critical.* Another woman said. *Because I can't deal with the heavy duty today. All I can deal with is the plain, brief.*

Information quality: Understanding the quality or authoritativeness of information is an issue for anyone seeking health information and was regarded as important by the women in the earlier focus groups. A 'quality report' is provided for each information item retrieved. The quality report is a brief summary of the major characteristics deemed to represent 'quality' and credibility. Each report notifies the user as to the creator of the material, the publisher status, the evidence base, as well as such features as 'purpose', whether the material contains references, and its currency. The women were presented with the statement 'I felt confident about the reliability and quality of the information provided', 10 of the women either agreed or agreed strongly with this statement, indicating the value of the report and the importance of knowing the quality of the resources. All users read the quality report, 8 before accessing the resource and 5 after.

The usability testing phase endorsed the value of including a 'quality report' alongside each retrieved resource, with users indicating that it assisted them in determining whether or not to retrieve a particular document, or in which order to view the material. *I guess it helped me prioritise the order in which I would look at them.* When asked what the best feature was one women explained:

"*The information provided with each found documents, i. e. on results of search page, description, quality and more information. This saves time and is very useful - don't need to look at every article found, better than 'google' info*".

The participants saw the quality report as valuable and enabled each user to extract and determine the specific criteria, which she deemed as a priority. For example, some users were reluctant to trust commercial sites, others felt that currency was of most importance. In providing the user with a comprehensible 'upfront' summary, these individual values were permitted their role in resource selection by the user as evidenced by their comments.

Country of origin: Given the fact that medical regimens and standards vary from country to country, the ability to locate material that is specifically geared to a 'local' audience was viewed as an important feature of any information provision strategy. In addition to the potential divergence in medical information, women in the initial focus groups repeatedly spoke of the importance of the accessibility of local support services, as well as facilitative information on various government subsidy schemes. The usability testing confirmed that users were appreciative of Australian content.

Meeting users' expectations: A critical question for the researchers was did the portal meet the expectations of women with breast cancer? Is it able to meet their needs? The women were asked during the focus groups if their expectations of the portal had been met. Generally the women were very enthusiastic and even though the portal contains only 500 resources the expectations of the women were met. For some as expressed by this woman it was because it was quick *I found a short cut to the information needed, and it worked.* Another women looking for information specifically to explain the disease to a child was very pleasantly surprised by the number of documents she was able to retrieve.

The data from the focus groups, questionnaire and numerous e-mail responses from women, their families and friends, have enabled the project team to address the concerns raised by the participants and many of the suggestions made by users were incorporated into the current design and layout of the portal interface.

Overall, the usability testing and evaluation confirmed that BCKOnline was positively received by women with breast cancer. Most importantly, the comments endorsed the overall objectives of the project, namely the provision of differentiated information, which may be selected on the basis of user preference and values.

19.7.4 Intelligent Characteristics in BCKOnline Portal

In the following section we discuss the intelligent features present in the BCKOnline portal, and indicate areas for further development based on the definition of how an information portal can function as an IDSS introduced earlier in this chapter.

19.7.4.1 Knowledge Repository and Memory

The BCKOnline knowledge repository comprises a virtual collection of information resources relating to breast cancer; user-centric descriptions of these resources including quality reports (contained in and managed by a metadata repository – see also below); links to the resources in the virtual collection; user profiles that enable the customized search strategy; and a capacity to match profiles selected by users with relevant resources. The portal captures expert knowledge through the selection, assessment and description of resources, and its user profiling. Some of the resources, especially those in the medical category, are selected and assessed on the basis of expert interpretation of evidence-based criteria available in the public domain, while the selection and quality assessment of other resources, including those relating to the experience of the disease and personal stories, requires the exercise of expert discretionary judgement. It is essential that the virtual collection of information resources to which the portal provides access is regularly reviewed and updated to reflect the state-of-the-art in terms of research about and treatment of breast cancer, as well as the personal experiences of women living with the disease, and support services and facilities, and that the information needs of the breast cancer community are monitored. Ongoing involvement of domain experts is essential to these task.

19.7.4.2 Intelligent Information Retrieval

With the aim of building the portal as an IDSS, it was imperative to adopt a cognitive rather than a computational information retrieval model, which maps with the portal's objective of providing differentiated access to resources. The portal therefore incorporates a context-based retrieval system, which matches diverse and dynamic user needs with relevant information resources/situated information. The aim is to retrieve resources that are highly relevant rather than to recall all relevant resources.

User profiles drawn from the user information needs analysis, were used as the means of building 'situation-dependent relevance', defined as a dynamic relationship between user needs and information resources, within the portal. The user's first interaction with the system involves selecting values that best match her circumstances to a predefined profile, *e. g.* age group, the stage of the disease, and information style preference (*e. g.* scientific, plain, brief or detailed). The system matches this input with the metadata descriptions of the resources and their target audience in order to establish whether relevant resources are available via the portal. For the purposes of the BCKOnline portal an operational definition of 'user profiling' relates to the capacity to generate a dynamic description of a particular user, referring to the predefined profiles, which in turn enables a metadata-driven search to pinpoint the most relevant resources and to provide the user with summary information about them, including a quality report based on expert assessment, to assist the user in deciding which resources are most relevant to her. The user-centered design of the portal interface and customized search strategies also enhance the portal's information retrieval capacity.

19.7.4.3 Adaptivity and Personalization

The personalization of BCKOnline portal is achieved through use of user-centric resource descriptors and profiling as described above, and through the quality reports that enable users to make decisions about the quality of the resource, *i. e.* their fitness for purpose, in terms of their own value systems. The BCKOnline portal is not adaptive by itself in that it does not learn about user needs via use of data-mining and related techniques to analyze previous information-seeking behavior and patterns of use. Adaptivity is currently implemented primarily through knowledge repository maintenance based on expert input and evaluation of portal usage. It is envisaged that statistical information will also be collected about the level of usage and needs for certain resources. Other learning mechanisms to ensure continuous growth of the system are being considered as part of a second phase of the project, *e. g.* the use of suitable technology such as text mining and supervised learning to collect automatically information about portal usage. Systematic analysis of this data could then be used to trigger further improvement of the content of the portal's knowledge base to suit user needs.

19.6.4.4 Information Classification and Prioritizing

The BCKOnline portal classifies information resources relating to breast cancer using the BCKOnline Metadata Schema and related encoding schemes that, as discussed earlier in the chapter, were developed and refined using the results of the user information needs analysis and expert-domain knowledge. At this stage, it does not employ text mining or ES approaches to assist in classifying resources.

Through the user information needs analysis it became apparent that women with breast cancer require different types of information to better inform their decision making. For example, making decisions relating to treatments, care, lifestyle and so on requires different types of information. Therefore the portal uses three main categories to describe the types of information available, classifying relevant resources in terms of whether they provide medical, personal or supportive information.

Medical - information about treatment and management of the disease, such as various treatment options; clinical trial reports, drug news.

Supportive - the effect of the disease on the woman and her family, aspects of social and psychological functioning, facilitative information that may include addresses of support groups, and government assistance information.

Personal - information based on experience of the disease including personal stories of other women.

For example, a user may want information on one of the cancer treatment drugs, tamoxifen. However, she could want *medical* style information explaining the purpose of the drug, or may be more interested in *personal* accounts of side effects, or *supportive* information about government subsidies for the drug. Resources are also classified, using the Metadata Schema and related encoding schemes, according to a number of other attributes, including their subject, target audience and quality. This type of context-rich, user-centric classification of resources used in the portal underpins the discovery and retrieval of highly relevant resources. At this stage relevance ranking based on text mining or computational techniques has not been incorporated, but it is being considered in the next phase.

19.7.4.5 Explanation Facility

The BCKOnline portal provides transparency through explanation of the kind of knowledge accessible and disclosure of resource selection criteria, the rules for resource description, and how quality is defined. Information about the reliability, quality and authoritativeness of the information resources available via the portal is provided to enable user assessment of the resources discovered based on fitness for purpose and personal values. The portal aims to empower users to make better-informed decisions by capturing and providing expert knowledge about the quality of the selected resources available through the portal. The user has an opportunity to see an explanation of the quality assessment of a resource as a part of the portal's functionality and before deciding to access the resource. Search output results include an option of displaying the Quality element and its qualifiers. They include the credentials of the creator/publisher/contributor, *e.g.* lay author(s), clinician(s), researcher(s), consumer group, commercial body/group, educational institution, government body, medical organization or cancer organization; information currency; the review process that the document was subjected to; whether the information is evidence based; its purpose, *e.g.* educational/informative, commercial, reportage of results, discussion forum; and the degree to which the resource represents a consensus view or is controversial in nature.

19.7.4.6 Metadata as a Mechanism for Providing User-centered Intelligent Decision Support

The BCKOnline portal consists of a metadata repository. The metadata descriptions captured and managed in the repository are created manually using the BCKOnline Metadata Schema adapted from the AGLS metadata schema (AGLS 1998), and related encoding schemes. During the first phase of the BCKOnline project, using a particularly resource intensive approach, has resulted in very high quality metadata. As yet the portal does not incorporate a metadata broker facility of the kind described earlier in the chapter. Further research is underway in a second phase of the project to enable the use of the intelligent features of ES to support the development of fuller metadata broker functionality in the portal, including tools for updating resource descriptions and automated assistance with the generation of metadata, based on expert-system approaches, agent-based text mining and supervised learning procedures

19.7.5 Extent to which BCKOnline Functions as IDSS

From the analysis presented it is clear that the BCKOnline portal exhibits major features that qualify it to be treated as an IDSS. Table 19.2 summarizes the ways IDSS functionality is implemented in the BCKOnline portal.

In summary, we examined an information portal as a gateway - a single point of access - to information resources. For effective information provision a portal as an integrated intelligent system is much more powerful than as a stand alone DSS or an ES system. As an IDSS, the portal needs to exhibit at a minimum level six core functionalities, including knowledge repository and memory, intelligent information retrieval, information classification and prioritization, explanation facility, personalization and adaptivity, and metadata brokerage. To illustrate this we considered how the BCKOnline portal could play the role of an IDSS. The BCKOnline portal design is based on extensive use of metadata for user-centric resource description and profiling. It implements a metadata-driven search, based on built-in knowledge about the problem domain encoded into a metadata schema in order to guide the users intelligently through the process of locating and delivering quality information.

BCKOnline, through its user profiles, metadata repository of user-centric resource descriptions, and extensions of approaches to describing target audiences and resource quality, attempts to address the diverse and changing information needs of the consumer. The preliminary usability results demonstrated increased satisfaction when users experienced the change from being passive recipients of an avalanche of information to active seekers of highly relevant information resources, ‘expert patients’ in charge of their health and wellbeing. This model could be more successful in the future world when the elements specified in metadata standards are encoded by the information publishers.

Table 19.2. IDSS functionality and how it has been implemented in BCKOnline portal

IDSS functionality	***BCKOnline functionality***	***Ways of implementation***
Knowledge repository and memory	Virtual collection of information resources, resource descriptions, quality reports, user profiles, links to resources described, capacity to match user profiles with relevant resources	Selection description and assessment of information resources to be made accessible via the portal, and development of user profiles on basis of expert-domain knowledge and user information needs analysis
Intelligent information retrieval	Customized search capability based on context-based retrieval system, which matches diverse and dynamic user information needs with relevant information resources/situated information	User profiles as means of providing 'situation dependent relevance'. Metadata-driven search strategies based on user profiling and user-centered resource description. User-centric interface design
Information classification	Categorization of all resources into three types of information – medical, supportive and personal – and classification of various attributes including subject, target audience and quality.	User-centered resource description based on BCKOnline Metadata Schema and related encoding schemes
Prioritizing	Not currently implemented	N/A
Explanation facility	Display of Quality element and its qualifiers; provision of information about selection criteria, types of resources included, basis of quality report and rules for description.	Documents available via portal detailing selection criteria, BCKOnline Metadata Schema, definition of quality criteria.
Personalization	Information retrieval based on user selected profiles, which match their circumstances, and provision of quality report that enables users to make judgments about quality for themselves.	User-centered resource description and profiling; and quality assessment based on expert-domain knowledge, and user information needs analysis
Adaptivity	Learning from usage patterns not currently implemented.	N/A
Metadata	Metadata repository to capture and manage standardized, user-centric resource descriptions	HotMeta, BCKOnline Metadata Schema and related encoding schemes.

19.8 Conclusion

The aim of this chapter was to broaden the perspective on intelligent decision support beyond particular technological approach, but rather consider it from the knowledge-based perspective. As a mechanism for knowledge production an intelligent portal plays the role of a gateway to externally stored information resources. It aims to satisfy the knowledge needs of the user -*e. g.* decision maker- by selecting quality information and providing it in an appropriate format. Such mechanism has not been previously considered in any type of decision support systems. On the other hand, as we demonstrated by the healthcare application described in this chapter this kind of facility is greatly appreciated and highly praised by the users. Although the outcomes of this study have been directed to supporting the breast cancer community, the study also demonstrates that they are generalizable and can be applied to other areas within the health domain and to online information provision more generally.

At the time when Internet searches become a major source for information in a broad range of decision situations, in which expert knowledge may not be available, it is important to consider building intelligence into the information resources. In extended intelligent decision support architecture we envisage decision support facilities that not only use internal data bases but fully exploit relevant external information accessible from the Internet. The interface of such architecture requires to possess the intelligence of an "electronic librarian", which takes charge of providing quality assurance in the information provision for decision support.

References

AGLS (1998) Australian Government Locator Service http://dublincore.org/archives/1999/dc7/AGLS.htm Accessed 09.03,2003.

AMA: American Medical Association (AMA) – Guidelines for medical and health information sites on the internet (2000) Document online at: www.ama-assn.org/ama/pub/category/1905.html

Angehrn A, Luthi S (1990). Intelligent decision support systems: a visual interactive approach. Interfaces; 20(6):17-28.

Asprey L, Middleton M (2003) Integrative Document and Content Management- Strategies for Exploiting Enterprise Knowledge, Idea Group Publishing

Bidgoli H. (1993), Integration of technologies: an ultimate decision-making aid, Journal of Industrial Management & Data Systems, 93:1:10-17.

Burstein F, Linger H (2003) Supporting post-Fordist work Practices: a Knowledge Management framework for Dynamic Intelligent Decision Support, Journal of IT&P special issue on KM, 16:3:289-305.

Burstein F, Fisher J, McKemmish S, Manaszewicz R, Malhotra P(2005) User centred quality health information provision: benefits and challenges, Hawaii 2005 Systems Sciences Conference, HICSS'38.

Burstein F, Smith H, Arnott D, O'Donnell P (1994) On the Nature of Intelligent Decision Support Systems , in Readings in DSS,Arnott & O'Donnell (eds), Monash University, Melbourne.

Christensen H, Griffiths K (2000) The Internet and mental health literacy. Australian and New Zealand Journal of Psychiatry 34: 975 - 979.

CRKM (2003-5) Clever Recordkeeping Metadata ARC Project, Create Once, Use Many Times: The Clever Use of Metadata, sponsored by NAA, State Records NSW and Australian Society of Archivists; Chief Investigators: McKemmish (Principal CI, Monash), Gilliland-Swetland (PI, UCLA) and Cunningham (PI, NAA), http://www.sims.monash.edu.au/research/rcrg/research/crm/index.html

DCMI (1995-2003) Dublin Core Metadata Initiative URL: http://dublincore.org/ Accessed 03.09.03.

Dhar V, Stein R. (1997) Intelligent decision support methods: The science of knowledge work, Prentice Hall.

Dumas J, Redish J. (1994) A Practical Guide to Usability Testing. Norwood, Ablex Publishing Corporation

Eysenbach G, Kohler C (2002) How do consumers search for and appraise health information on the world wide web? Qualitative study using focus groups, usability tests, and in-depth interviews. British Medical Journal, 324(7337):573-579

Firestone JM (2003) Enterprise Information portals and knowledge management, BH.

Ford N, Miller D, Moss N (2002) Web search strategies and retrieval effectiveness: an empirical study, Journal of Documentation, 58:1:30-48

Gregor RM (1991) The evolving technology of classification-based knowledge representation systems , In John F. Sowa (Eds), Principles of Semantic Networks, (1st edn), Morgan Kaufmann Publishers, San Mateo, California.

Gregor S, Benbasat I (1999) Explanations from intelligent systems: Theoretical foundations and implications for practice , MIS Quarterly, 23:4: 497-530

Hackos J, Redish J (1998) User and Task Analysis for Interface Design. NY, Wiley&Sons.

Harvard Computing Group (2002). Document on line at: www.harvardcomputing.com

Health Online: A Health Information Action Plan for Australia (2004) http://www.health.gov.au/healthonline/about/about.html (Accessed 10/9/04) Commonwealth Department of Health and Ageing

Hi-Ethics (2004) Website: http://www.hiethics.com

Hollnagel E (1986) Cognitive system performance analysis, In E. Hollnagel, G. Mancini and D.D. Woods (Eds), Intelligent Decision Support in Process Environments, (1st edn), NATO ASI Series

Hollnagel E, Woods DD (1983) Cognitive systems engineering: new wine in new bottles, International Journal of Man-Machine Studies, 18:6:583-600

Holsapple C, Whinston A (1996) Decision Support Systems: A Knowledge Based Approach. Minneapolis, MN: West Publishing, Inc.

HotMeta (2004) Metadata Search Engine website http://www.dstc.edu.au/Research/Projects/hotmeta/ last accessed June 2004.

Jansen BJ, Pooch U (2001) A review of web searching studies and a framework for future research, Journal of the American Society for Information Science and Technology, 52:3:235-246

Jones K (1999) Linguistic searching versus relevance ranking: DR-LINK and TARGET , The Electronic Library, Online & CD-ROM review, 23:2:67-80

Klein MR, Methlie LB (1995) Knowledge-based Decision Support Systems, Wiley&Sons.

Linger H, Burstein F (1997) Intelligent DSS in the Context of Modern Organisation, Proceedings of the 4th Conference of the International Society for Decision Support Systems, 429-444.

McKemmish S, Acland G, Ward N, Reed B (1999) Describing records in context in the continuum: the Australian Recordkeeping Metadata Schema, Archivaria, 48:3-43

Nie J, Lepage F (1998), Toward a broader logical model for information retrieval . In Fabio Crestani, Mounia Lalmas and Cornelis Joost van Rijsbergen (Eds), Information retrieval: Uncertainty and Logics, (1st edn), Kluwer.

Nielsen J,d Molich R. (1990) Heuristic evaluation of user interfaces. Proc. ACM CHI'90 Conf. (Seattle, WA, 1-5 April): 249-256.

Norman DA (1986), New views of information processing: implications for intelligent decision support systems. In E. Hollnagel, G. Mancini and D.D. Woods (Eds), Intelligent Decision Support in Process Environments, (1st edn), NATO ASI Series.

Power DJ (2002), Decision Support Systems: Concepts and Resources for Managers, Westport, CT: Greenwood/Quorum Books, 2002.

Raggad BG, Gargano ML (1999), Expert system: defection and perfection, Logistics Information Management, 12:5:95-406

Renato I, Waugh A (1997) Metadata: Enabling the Internet, CAUSE97 Conference, Apr. 97, [on line] http://www.dstc.edu.au/RDU/publications/cause97/ last accessed 06/2004.

Rijsbergen CJ Van (1979), Information Retrieval , Butterworth-Heinemann

Schmidt K, Ernst E (2004). Assessing websites on complementary and alternative medicine for cancer, Annals of Oncology, 15: 733-742

Shilakes C, Tylman J (1998), Enterprise information portals, NY: Merrill Lynch

Spink A, Jansen BF, Ozmultu HC (2000) Use of query reformulation and relevance feedback by excite users, Electronic Networking Applications and Policy, 10:4:317-328

Sprague RH, Carlson ED (1982) Building Effective Decision Support Systems, Englewood Clifts, N.J.: Prentice-Hall, Inc.: 1982.

Thewes B, Butow P,. Girgis A, Pendlebury S (2003). The psychosocial needs of breast cancer survivors; A qualitative study of the shared and unique needs of younger versus older survivors. Psycho-Oncology 13(3): 177-189

Turban E, Aronson JE (1995) Decision Support and Intelligent Systems (5th edition), Upper Saddle River, N.J.: Prentice-Hall, Inc.: 1995.

Turban E, Watkins PR (1986), Integrating expert systems and decision support systems, MIS Quarterly, 10:2:121-134

Williamson K, Manaszewicz R (2003) Breast Cancer Information Needs and Seeking: Towards an intelligent, user sensitive portal to Breast Cancer Knowledge Online , The New Review of Information Behaviour Research, 3, Cambridge: Taylor Graham Publishing, 48 Regent St, Cambridge CB2 1FD.

20

An Overview of Future Challenges of Decision Support Technologies

Nikos Karacapilidis

Industrial Management and Information Systems Lab, MEAD, University of Patras,
26504 Rio Patras, Greece, e-mail: nikos@mech.upatras.gr

Arguing that contemporary decision making needs to be tackled through a holistic perspective, in that the conceptual, methodological and application-oriented aspects of the problem have to be simultaneously taken into account, this chapter provides an overview of challenges for the future development of decision support technologies and their integration in intelligent decision-making support systems. Based on this discussion, and aiming at providing decision makers around the world with applications of enhanced performance, while, at the same time, addressing their communication and collaboration needs in an efficient and effective way, an advanced Web-based decision making framework is proposed.

20.1 Introduction

Decision making is ubiquitous in the contemporary organizational processes. According to Simon (1977), it comprises three principal phases: identifying problematic situations or opportunities that call for decisions (*intelligence* phase), inventing or developing possible courses of action and testing of their feasibility (*design* phase), and selecting a certain course of action to be followed (*choice* phase). As noted in (McLauglin 1995), "*successful organizations outdecide their competitors in at least three ways: they make better decisions, they make decisions faster, and they implement more decisions*". The quality, speed and realization of the decision making can be increased when the right information is available to the right persons, at the right time and in the right form. However, especially in the context of ill-structured problems, scientific decision making is a highly complex task, characterized by conditions such as (Karacapilidis and Pappis 1997):

- The coexistence of not enough and too much information: For some parts of the problem, relevant information that would be useful for making a decision may be missing or insufficient, whereas for other parts the time needed for the retrieval and comprehension of the existing information volume may be prohibitive. Regarding the efficiency of the system, response time is often a basic issue. Moreover, independent of how much information is available, decision-makers' opinions may differ about its truth, relevance or value for deciding an issue.

- The decision-making procedure is often performed through a lot of debates and negotiations among a group of people. Decision makers may have arguments in favor of or against alternative courses of action. Similarly, they may adopt and, consequently, suggest their own strategies that fulfill some goals at a specific level. Conflicts of interest are inevitable and support for achieving consensus and compromise is required.

- Reasoning is defeasible, *i. e.* further information can trigger another alternative to appear preferable to what seems best at the moment.

- Factual knowledge is not always sufficient for making a decision; value judgments, depending on the background, personality type, cognition, decision style, company position, and objectives of each decision maker, are often the most critical issues.

- Last but not least, decision makers are not proficient in mathematics or computer science. The system should provide them with the appropriate tools in order to participate in the discussion in a "natural" way. This is in accordance with the vision of pioneers in the decision support systems field, that is, by supporting and not replacing human judgment, the system comes in second and the users first.

Systems aiming at offering computerized support in the decision making process first appeared in the late 1960s. *DecisionsSupport systems* (DSS) have been defined as "interactive computer-based systems, which help decision makers utilize data and models to solve unstructured problems" (Gorry and Scott-Morton 1971). According to Keen and Scott-Morton (1978), DSS "couple the intellectual resources of individuals with the capabilities of the computer to improve the quality of decisions". As pointed out in (Pearson and Shim 1995), research on this field during the past three decades has mainly focused on how information technology can improve the efficiency with which a user makes a decision, and can improve the effectiveness of that decision. More specifically, quoting (Shim *et al.* 2002), much research and practical effort has been conducted in developing technologies to be exploited in DSS components for "sophisticated database management capabilities with access to internal and external data, information and knowledge, powerful modeling functions accessed by a model management system, and powerful, yet simple user-interface designs that enable interactive queries, reporting and graphing functions".

The main purpose of this chapter is to provide an overview of challenges for the future development of *Decision support technologies* (DST) and their integration in *Intelligent decision-making support systems* (i-DMSS). Towards this aim, we have conducted a survey of the relevant literature to reveal prominent technologies and point out issues to be further investigated by the DMSS research community.

Accordingly, we have developed an advanced Web-based decision-making framework that builds on these technologies to thoroughly address the above issues. The ultimate objective of the proposed framework is to provide geographically dispersed decision makers with applications of enhanced performance while simultaneously addressing their communication and collaboration needs in an efficient and effective way.

The remainder of this chapter is structured as follows: The next section discusses background issues and comments on the evolution of decision support technologies. Section 20.3 focuses on the most prominent of them and outlines their advantages and disadvantages. Section 20.4 describes in detail the proposed decision-making framework and highlights DST development issues. Finally, Section 20.5 concludes by discussing the potential contribution of the proposed framework for the iDMSS field.

20.2 Background Issues

Generally speaking, technologies for decision support are assembled from four basic components, namely *data*, *models*, *knowledge*, and *user interface* (Turban and Aronson 2001, Mora *et al.* 2003). It is the particular assemblage of the above components that defines the features and the functionality of a DSS (these components are related to the subsystems a DSS consists of). More specifically, the first of them is associated with the *data- management subsystem*, which handles the extracting, organizing and archiving an organization's internal and external data. Development of the appropriate database, database management system (DBMS), data directory, and query facility are the major issues to be addressed in this subsystem. The second component relates to the *model-management subsystem*, which provides the DSS's analytical capabilities. Development of a quantitative model base and tools to manage the creation, updating, integration and execution of its constituent models are the critical issues here. In turn, the knowledge component is associated with the *knowledge-based management subsystem* that characterizes more advanced DSS, and provides the knowledge and expertise needed to solve aspects of an ill-structured problem. Such an aid may concern issues related to the decision process *per se* (steps to be followed), the models to be built, and the manipulation of the uncertainty inherent in the problem. Finally, the last component is associated with the *user-interface subsystem*, which covers all aspects of communication and interaction between a user and the DSS.

In the 1970s and early 1980s, DST were customarily focused on model development and problem analysis, while over the last two decades the related research has evolved to include additional concepts and views (Forgionne *et al.* 2002, Shim *et al.* 2002). These include *Group decision support systems* (GDSS), attempting to provide evaluation of ideas in a brainstorming setting as well as to facilitate communications between remote users, *Executive information systems* (EIS), extending the scope of a DSS from personal or small group use to a corporate level, and *Knowledge-based DSS*, aiming at offering enhanced support to decision makers by encapsulating techniques from the disciplines of *Artificial intelligence* (AI).

The advent of the Internet and the Web, as well as modern communication technology has resulted to the broadening of the organizational environment. Courtney (2001) suggested that DSS researchers have to embrace a much more comprehensive view of the organizational decision-making context and accordingly develop systems that are able to handle "softer" information. What started to evolve in the last few years is that issues related to the mental models of decision makers, expressing their organizational, personal and technical perspectives of the problem under consideration, are critical and have to be carefully addressed.

Summarizing, it is clear that the introduction of DSS received great attention from the beginning, since these systems were heading to important developments such as the integration of interactive systems for managers and professionals, the achievement of user-friendly environments, and the provision of a suitable framework for the handling of semistructured and unstructured tasks. However, we argue that research on this area, having over-dealt with technological and definition issues (*e. g.* the differences between a DSS and an *Expert system* or an *Executive information system*), has de-emphasized other major issues in improving decision making (Alter 1992). These issues include work structuring in order to improve coordination, use of communication technology to make decision making more efficient and effective, enforcing of rules and procedures for achieving consistency, and (semi)automation of data processing in data-intensive decision making situations. Research on DSS design has only recently acquired a strong organizational focus (Zuurbier 1992). Ten years ago, Angehrn and Jelassi (1994) urged the DSS community to further consider the conceptual, methodological and application-oriented aspects of the problem. *Conceptual focus* is associated with the consideration of the nature of individual and organizational decision-making processes, *methodological focus* with the integration of existing computer-based tools, techniques and systems into the human decision-making context, and *application-oriented focus* with the consideration of the real organizational needs by extending decision support to business teams. Considering the above aspects, a series of prominent technologies has been proposed and evolved. These are presented in the next section, which starts shaping the proposed decision-making framework.

20.3 Prominent Decision Support Technologies

Data warehouses, online analytical processing, data mining and Web-based DSS have been broadly recognized as technologies playing a prominent role in the development of current and future DSS (Shim *et al.* 2002, Turban and Aronson,2001). More specifically, *data warehouses* provide the infrastructure that enables businesses to extract, cleanse, and store vast amounts of corporate data from operational systems for efficient and accurate responses to user queries (Inmon 1996, Kimball and Ross 2002) and empower knowledge workers with information that allows them to make decisions based on a solid foundation of facts (Devlin 1997). However, only a part of the required knowledge can be represented in computers; a data warehouse does not provide adequate support for knowledge intensive queries in the organization. As argued in (Nemati *et al.* 2002), what is needed is "a new generation of knowledge-enabled systems that provides the infrastructure required to capture, enhance, store, organize, leverage, analyze, and disseminate not only data and information but also knowledge".

Data stored in a data warehouse are usually analyzed with the aid of *on-line analytical processing* (OLAP) tools (Berson and Smith 1997, Thomsen 2002). OLAP has been defined as "a category of software technology that enables analysts, managers and executives to gain insight into data through fast, consistent, interactive access to a wide variety of possible views of information that has been transformed from raw data to reflect the real dimensionality of the enterprise as understood by the user" (Power 1999). According to the database technology for building a data warehouse, two basic types of OLAP tools are distinguished, namely *Multidimensional OLAP* (MOLAP) and *Relational OLAP* (ROLAP). Each of these types has its own advantages and disadvantages, while a third one, namely *Hybrid OLAP* (HOLAP), attempts to combine the advantages of the first two. More specifically, MOLAP is the more traditional type of OLAP analysis, where data is stored in a multidimensional cube. It is more appropriate for cubes with frequent use and when there exists a necessity for rapid query response. It can quickly perform complex calculations, since all of them have been pregenerated (at the creation of the cube). However, it is often limited in the amount of data it can handle (this is related to the amount of data that can be included in the cube), and requires additional investment from an organization. On the other hand, ROLAP performs dynamic analysis of data stored in a relational database and leverages its functionalities. It does not use precalculated data cubes; instead, it intercepts the query and poses the question to the standard relational database and its tables in order to bring back the data required to answer the question. Its disadvantages are slow response and limited scalability (depending on the technology architecture that is utilized). However, compared to MOLAP, it supports larger user groups and greater amounts of data and is often used when these capacities are crucial.

The power of the above applications in processing vast amounts of data can be further augmented by *data-mining* applications. Such tools are built on concepts and techniques from AI and Statistics (such as Case-based reasoning, data visualization, fuzzy analysis, and neural networks), aiming at providing a more sophisticated data analysis by discovering patterns of data and inferring data content relationships and rules from them (Fayyad *et al.* 1996, Berson and Smith 1997). Contemporary data mining applications can learn from the previous history of the investigated system, as well as shape and test hypotheses about the rules that this system acts upon. When that appropriate knowledge has been formulated, they can be incorporated into a DSS to aid managers make better decisions. The ever-growing body of information that exists in the World Wide Web can also be exploited through the use of the above technologies to support a series of decision making settings. As argued in (Han and Chang 2002), "*the Web, an immense and dynamic collection of pages that includes countless hyperlinks and huge volumes of access and usage information, provides a rich and unprecedented data mining source*".

At the same time, the Web environment becomes a widely adopted development and delivery platform. *Web-Based DSS* deliver information and/or tools to a decision maker through a Web browser that is accessing the Internet or a corporate intranet. The computer server that is hosting the DSS application is linked to the user's computer by a network with the TCP/IP protocol. Web-Based DSS can be communications, data, document, knowledge, or model driven, or built following a hybrid approach (Power 1999). Depending on the network type they are based on, they can provide specific decision making capabilities to diverse user types, such as to managers over an intranet, customers and suppliers over an extranet, or to any stakeholder over the Internet (Shim *et al.* 2002).

Two additional DSS technologies, falling into the Artificial Intelligence (AI) discipline, are *Rule-based systems* (Hayes-Roth 1985, Ignizio 1990) and *Case-based reasoning* (Kolodner 1993, Watson 1997). *Rule-based systems* (RBS) do not represent knowledge in a declarative and static way; instead, they do so through a set of "if-then" rules that indicate what has to be done or concluded at a specific instance of the problem under consideration (*i. e.* given a set of facts). Reasoning is performed through either *forward chaining*, where using the initial facts the system exploits rules to draw new conclusions or take certain actions (data-driven approach), or *backward chaining*, where the system attempts to satisfy the goals in the goal stack by finding rules that can conclude the information needed and trying to satisfy the "if" parts of those rules (goal-driven approach). Generally speaking, RBS are of practical importance for problems for which the related knowledge can be expressed in the form of the above rules and the problem area is not large. If there are too many rules, RBS become difficult to maintain and are characterized by low performance.

On the other hand, according to the *Case-Based Reasoning* (CBR) technology, expertise is encoded in a library of past cases (not in rules). Typically, each case comprises a description of a certain instance of the problem and its solution or outcome (the knowledge and the reasoning process followed to reach the solution is not explicitly recorded). To solve a new problem instance, a matching of it against past cases is performed (according to diverse similarity measures), the aim being to retrieve similar cases and exploit their solutions. These solutions may be revised for the new instance of the problem, while the new instance and its final solution shape a new case to be stored in the case base. CBR received growing interest in the last decade, both from an academic and commercial point of view. Its suitability to a decision-making context depends on a set of parameters related to whether records of previously solved problem instances exist, historical cases are viewed as a valuable asset that has to be retained, and exploitation of previous experiences is considered as useful and is common practice.

All the above technologies certainly facilitate diverse aspects of decision making. Although there exist certain limitations in their suitability, they may aid DSS users to make better and faster decisions. However, we argue that there is room for further developing the conceptual, methodological and application-oriented aspects of the problem (Angehrn and Jelassi 1994). At the same time, what is still missing is a holistic perspective (Ackoff 1999). These are basically due to the growing need to develop applications by following a more *human-centric* (not problem-centric) view, in order to appropriately address the requirements of the contemporary, knowledge-intensive organization's employees. Such requirements stem from the fact that decision making has also to be considered as a *social process* that principally involves human interaction (Smoliar 2003). The structuring and management of this interaction requires the appropriate technological support and has to be explicitly embedded in the system. The above requirements, together with the ones imposed by the way decision makers work and collaborate today, delineate a set of challenges for further DST development, which are discussed in the following section and shape our view towards the development of an advanced Web-based decision-making framework.

20.4 Future Challenges and the Proposed Decision-making Framework

The evolution of telecommunications network technology has dramatically facilitated the sharing of information and the participation of individuals in the decision making process. Group decision making becomes a necessity in the contemporary enterprise (Fjermestad and Hiltz 2000); the more different perspectives are taken into account, the smaller the chances of addressing the wrong problem and reaching an inadequate solution (Vennix 1996). *Group decision support systems* (GDSS) have been defined as interactive computer-based systems that facilitate the solution of ill-structured problems by a set of decision makers that work together as a team (Kreamer and King 1988). The main objective of a GDSS is to augment the effectiveness of decision groups through the interactive sharing of information between the group members and the computer (Huber 1984). This can be achieved by removing communication impediments, providing techniques for structuring decision analysis and systematically directing the pattern, timing, or content of the discussion (DeSanctis and Gallupe 1987). The environment in which the group decision making procedure takes place sets different communication requirements and defines alternative types of GDSS. Alternative taxonomy schemes for these systems, justified across various dimensions of design issues, have been proposed in the literature (Jarke 1986, Jelassi and Foroughi 1989).

At the same time, as argued in (Prahalad and Hamel 1990), "*a firm's only advantage in today's business environment is its ability to leverage and utilize its knowledge*". While a firm comprises individuals and a set of definable objectified resources, its most strategically important feature is its body of *collective knowledge* (Spender 1996). Such knowledge resides in an evolving set of assets including the employees, structure, culture and processes of the organization. Of these, employee knowledge, and particularly tacit knowledge is identified as the dominant one, which is decisive at all mental levels and has to be fully exploited (Nonaka 1994). Such an exploitation refers to the transformation of tacit knowledge to codified information, which is considered as a core process for economic activity and development (Cohendet and Steinmueller 2000).

The above advocate the adoption of a *knowledge-based decision-making view* (Holsapple and Whinston 1996), which should delineate the future development of decision support technologies. According to this view, decisions should be considered as pieces of descriptive or procedural knowledge referring to an action commitment. In such a way, the decision making process is able to produce new knowledge, such as evidence justifying or challenging an alternative or practices to be followed or avoided after the evaluation of a decision, thus providing a refined understanding of the problem. On the other hand, in a decision making context the knowledge base of facts and routines alters, since it has to reflect the ever-changing external environment and internal structures of the organization (Bhatt and Zaveri 2002). *Knowledge-management* activities such as knowledge elicitation, representation and distribution influence the creation of the decision models to be adopted, thus enhancing the decision-making process (Bolloju *et al.* 2002).

The above-mentioned synergy of decision making and knowledge management can be further strengthened by the incorporation of features enabling decision makers to perform argumentation and experimentations on the issues raised. Many collaborative decision making problems have to be solved through *dialoguing* and *argumentation* among a group of people (Toulmin 1958, Sycara 1990, van Eemeren *et al.* 1996, Walton 1996, Bell 1997, Provis 2004, Katzav and Reed 2004). In such contexts, conflicts of interest are unavoidable and support for achieving consensus and compromise is required. Each decision maker may formulate and put forward his/her own position that fulfills some goals with a specific acceptance level. Moreover, he/she may have arguments in favor of or against alternative solutions, as well as preferences and constraints imposed on them. Depending on the role and the goals of each decision maker, subjective estimates of the problem should be taken into consideration. Independently of the model used for decision making, argumentation is valuable in shaping a common understanding of the problem. It can provide the means to decide which parts of the information brought up by the decision makers will finally be the input to the model used. Moreover, as indicated in (Karacapilidis and Papadias 2001), argumentation may stimulate the participation of decision makers and encourage constructive criticism. To address the above category of requirements, a user-friendly discourse-based decision support environment should be developed.

On the other hand, controlled *experimentation by simulation* may further augment the decision-making process by providing insight into the dynamic interactions and feedback loops formed by the problem elements (Sierhuis and Selvin 1996, Sterman 2000, Miller *et al.* 2001, Taylor 2001). Being seamlessly integrated into a discourse-based decision support environment, a simulation model can map organizational knowledge onto appropriate graphs quantifying the problem under consideration, thus providing a clearer understanding of which alternative solution seems to be more prominent at the moment. Moreover, it can provide the means for an individual to conceptually define his/her position and perform experiments before asserting it to the dialoguing and decision support environment. Taking into account the current state of a discourse organized in an intelligent way, individuals may thoroughly contemplate their next move to assure that it will have the best impact on the ongoing discussion.

In any case, the efficient exchange of knowledge amongst the decision makers and thus the facilitation of their communication should rely on the establishment of a common language (terms of reference), as far as the representation of the issue, the assessment of the current situation and the objectives to be attained are concerned. The use of *ontologies* is valuable for such purposes (Chandrasekaran *et al.* 1999). From an information science point of view, ontologies are the hierarchical structures of knowledge about things, by subcategorizing them according to their essential or relevant cognitive qualities (Genesereth and Nilsson 1987). They are a means to accomplish a shared understanding of different knowledge domains and allow for sharing and reuse of bodies of knowledge across groups and applications (Duineveld *et al.* 2000). Moreover, they figure prominently in the emerging *Semantic Web* as a way of representing the semantics of documents and enabling these semantics to be used by web applications (Davies *et al.* 2003, Daconta *et al.* 2003). The challenges imposed through the establishment of an appropriate ontology schema in a decision making environment should be viewed together with the exploitation of the prominent technologies discussed in the previous section. It is expected that the use of ontologies will result in building more intelligent applications, enabling them to work more accurately at the humans' conceptual level.

In addition to the above, the integration of intelligent tools in a DSS can further improve its efficiency and effectiveness. Such tools build on the concept of *intelligent agents*, which are software entities that perform a set of operations on behalf of a user (or another program), thus acting as his/her personal assistant. Intelligent agents are personalized through the maintenance of each user's profile, and may accordingly perceive conditions holding in a dynamic environment, act with respect to these conditions, and reason to draw inferences and solve problems (Maes 1994, Weiss 1999, Cuena and Ossowski 1999, Wooldridge 2002). Their basic characteristics, namely, autonomy, proactiveness and intelligence, together with their ability to cooperate, make them suitable for the delegation of diverse decision making tasks, such as information and knowledge seeking, filtering and retrieval, monitoring of the decision making context under consideration, comparison and evaluation of alternative solutions, and negotiation among users of opposing interest.

In the context of electronic work (e-Work) in a decision-making environment, another challenge for DST is related to providing *customized solutions* that adapt to the decision maker's profile, taking into account one's preferences, abilities, experience, collaboration mode, as well as aspects related to technical specifications of his/her platform, software availability, and network connection (Kobsa 2001, Thomson *et al.* 2004). In order to be effective, such solutions have to remove barriers imposed by noninteroperable collaboration tools, inadequate infrastructure, undefined data sharing policies and standards, differing priorities for presentation formats, information that is not tailored to the end-user's environment, lack of confidence, and definition of roles and responsibilities. Issues to be addressed include:

- provision for personalized (based on *adaptive learning* techniques) collaboration tools that track a decision-maker's activity and interactions during experimentation with the system, analyze the feedback, and accordingly identify his/her needs or interests (Langley 1999, Churchill *et al.* 2001, Kobsa 2002);

- generation of *customized content* through approaches such as document transformation, dynamic documents generation, and adaptive hypermedia;

- the *adaptation of the quality of services* offered according to the available bandwidth and networked infrastructures, and

- *device independency* to face the variety of emerging devices and specific operating systems.

The above discussion shapes our view towards the development of an advanced Web-based decision-making framework that is able to satisfy the requirements of decision makers in the contemporary organization. As illustrated in Figure 20.1, we view decision making to be seamlessly integrated and highly interrelated with the processes of knowledge management, dialoguing and argumentation, and experimentation by simulation. All these processes are continuously supported by technologies that accurately address issues related to ontology management, intelligent agents, data warehouses, data mining, OLAP, case-based and rule-based reasoning, adaptive interfaces, and user modeling. The framework maintains a knowledge base, where data and knowledge related to the problem is maintained, and a model base with a library of decision-making and simulation models. Decision makers participate in an argumentative discourse-based decision-making process, through appropriately designed Web interfaces serving data and knowledge acquisition, loading, evaluation, and refinement purposes.

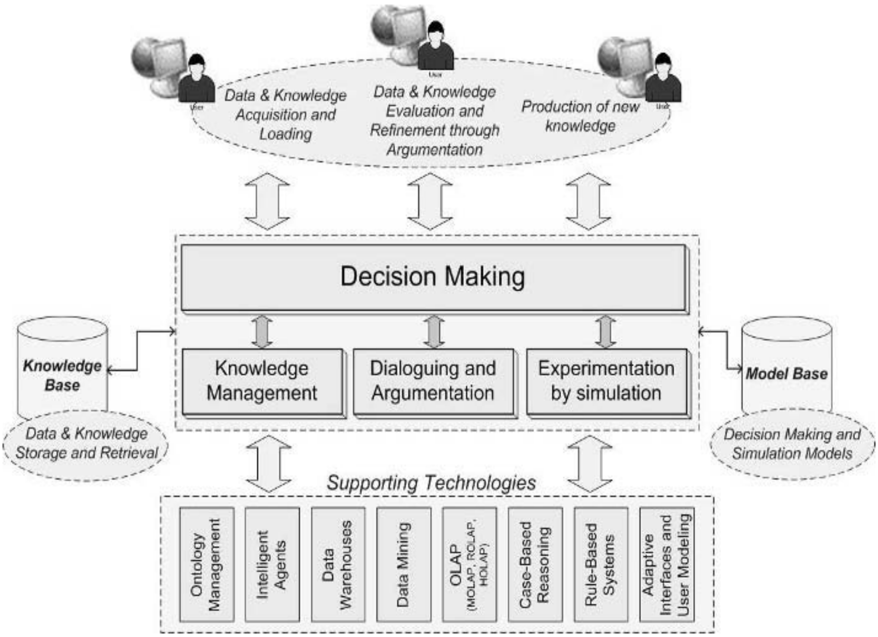


Figure 20.1. The proposed decision-making framework

Consider, for instance, a decision-making setting where a set of managers representing diverse divisions and business units of a multinational company has to make a decision about the placement of a new factory. Exploiting the proposed framework ("dialoguing and argumentation" component), these managers will be able to express their opinions (alternative locations), share them easily with their peers, and argue extensively on them. Such an interaction leads to constructive criticism, externalization of individuals' tacit knowledge, integration of it with problem-specific explicit knowledge and, consequently, organizational knowledge building on the problem under consideration ("knowledge management" component). More insights into the overall decision-making process can be gained through a simulation tool; feeding such a tool with input extracted from the positions stated (*e. g.* positions regarding a particular location) will result in a set of simulation results (*e. g.* concerning cost and time issues) that can further advance the above interaction ("experimentation by simulation" component). At the same time, decision makers can exploit the supporting technologies shown at the bottom part of Figure 20.1 to better support their positions and arguments. For instance, they can retrieve parts of past discussions (using data mining techniques and exploiting a well-structured ontology model) concerning similar issues (*e. g.* cost and time issues about a particular business unit) and "attach" them to their positions, thus validating them further. Moreover, they can analyze the results obtained from the simulation tool and view them through different perspectives with the aid of OLAP tools. Finally, the overall decision-making process concerning the selection of the most appropriate location for the placement of the new factory, beyond argumentation-based reasoning, can be also based on previous, well recorded and broadly accepted rules and cases.

20.5 Discussion and Conclusions

Decision support technologies should further exploit the relentless advances in computers and telecommunication infrastructure, the aim being to deliver applications of enhanced performance to decision makers around the world, while efficiently and effectively addressing communication and collaboration issues. Mining of data warehouses and use of analytical tools already provide decision makers with accurate information. However, this is not sufficient; the retrieved information has to be appropriately exploited in developing organizational memory, a process that, beyond storing individual and collective knowledge, is related to organizational learning, decision making, and competitive capability issues. The dynamic nature of knowledge has also to be emphasized, by taking into account individual and organizational decision-making perspectives.

Generally speaking, problems to be addressed through collaboration between geographically dispersed decision makers lack a unique, agreed-upon formulation or well-developed plans of action. Moreover, such problems could not be usually solved by formal models or methodologies. Instead, an argumentative practical reasoning approach seems to be the appropriate solution (Girle *et al.* 2003); as argued in (Buckingham Shum 2003), "an open-ended, dialectic process of collaboratively defining and debating issues is a powerful way of discovering the structure of such problems". What actually happens in the context under consideration is that all decision makers involved initially identify the main problems and issues to be addressed, and then propose possible actions and solutions. Next, for each of these actions and solutions, they articulate advantages and disadvantages according to their views and perceptions, and bring forward (in a direct or indirect way) preferences, which reflect their values, interests and expectations. Thus, collaborative decision making processes have a *rationality-related dimension* and a *social dimension*.

All the above issues have been thoroughly considered in the proposed decision making framework. Its main contribution is that it builds on a holistic approach. According to it, *decision support technologies have to be evolved and interrelated* in order to efficiently and effectively address the requirements of the knowledge-intensive organization. This evolution and interrelation aims at developing a more human-centric view of the problem, which appropriately structures and manages the underlying human interaction. The prominent decision support technologies discussed previously in this chapter should be exploited in any instance of the proposed discourse-based decision-making process in order to retrieve useful information and knowledge, as well as to reason according to previous cases or predefined rules. Establishment of the appropriate ontology schemas is crucial at this point. Integration of argumentation and experimentation features throughout the decision-making process is also a promising research direction. The proposed interdisciplinary approach should be interweaved with intelligent agent technologies, which are able to facilitate a variety of decision-makers' tasks and actions by acting on their behalf, as well as to automate the system's processes, such as selection of the suitable simulation or decision-making model for the problem under consideration and search for relevant information. Last but not least, much attention should be paid to the provision of customized user interfaces.

Summarizing, the proposed framework advocates wider and a more profound support in decision-making processes. This will be achieved by the joint consideration of organizational and technical issues, and accordingly, the seamless integration of technologies originally developed under the DMSS and AI fields. We argue that research work towards this direction will bridge the gap between the approaches developed from each side, in that they will both aim at augmenting the intelligence and support required in the complex and dynamic decision making processes of the contemporary organization.

Acknowledgements

The author thanks Prof. Costas Pappis for his valuable comments during the preparation of this chapter. Also, the anonymous referees for their suggestions for improvement of an earlier version of it.

References

Ackoff RL (1999) Re-creating the Corporation: A Design of Organizations for the 21st Century. Oxford University Press, New York.

Alter S (1992) Why Persist with DSS when the Real Issue is Improving Decision Making? In: Jelassi T, Klein MR, Mayon-White WM (Eds.) Decision Support Systems: Experiences and Expectations, IFIP, North-Holland, Amsterdam, The Netherlands, pp 1-11.

Angehrn A, Jelassi T (1994) DSS Research and Practice in Perspective. Decision Support Systems 12:267-275.

Bell P (1997) Using argument representations to make thinking visible for individuals and groups. In: Hall R, Miyake N, Enyedy N (eds.) Proceedings of CSCL'97: The Second International Conference on Computer Support for Collaborative Learning, University of Toronto Press, Toronto, Canada, pp 10-19.

Berson A, Smith S (1997) Data Warehousing, Data Mining, and OLAP. McGraw-Hill, New York.

Bhatt G, Zaveri J (2002) The Enabling Role of Decision Support Systems in Organizational Learning. Decision Support Systems 32(3):297-309.

Bolloju N, Khalifa M, Turban E (2002) Integrating Knowledge Management into Enterprise Environments for the Next Generation Decision Support. Decision Support Systems 33:163-176.

Buckingham Shum S (2003) The Roots of Computer Supported Argument Visualization. In: Kirschner P, Buckingham Shum S, Carr C (eds.) Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-Making. Springer Verlag, London, UK, pp 3-24.

Chandrasekaran B, Josepheson J, Benjamins VR (1999) Ontologies: What Are They? Why Do We Need Them? IEEE Intelligent Systems 14(1):20-26.

Churchill EF, Snowdon DN, Munro AJ (2001) Collaborative Virtual Environments - Digital Places and Spaces for Virtual Interaction. Springer Verlag, London, UK.

Cohendet P, Steinmueller WE (2000). The Codification of Knowledge: a Conceptual and Empirical Exploration. Industrial and Corporate Change 9(2):195-209.

Courtney J (2001) Decision Making and Knowledge Management in Inquiring Organizations: Toward a New Decision Making Paradigm for DSS. Decision Support Systems 31:17-38.

Cuena J, Ossowski S (1999) Distributed Models for Decision Support. In: Weiss G (ed.) Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. MIT Press, Cambridge, MA, pp 459-504.

Daconta, M.C., Obrst, L.J. and Smith, K.T. (2003). The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management. Wiley Publishing, Indianapolis, IN.

Davies J, Fensel D, van Harmelen F (2003) Towards the Semantic Web: Ontology-Driven Knowledge Management. John Wiley & Sons, Chichester, UK.

DeSanctis G, Gallupe RB (1987) A Foundation for the Study of Group Decision Support Systems. Management Science 33(5):589-609.

Devlin B (1997) Data Warehouse: From Architecture to Implementation. Addison-Wesley, Menlo Park, CA.

Duineveld AJ, Stoter R, Weiden MR, Kenepa B, Benjamins VR (2000) WonderTools? A Comparative Study of Ontological Engineering Tools. International Journal of Human-Computer Studies 52:1111-1133.

Fayyad UM, Piatetsky-Shapiro G, Smyth P, Uthurusamy R (1996) Advances in Knowledge Discovery and Data Mining. AAAI Press, Menlo Park, CA.

Fjermestad J, Hiltz SR (2000) Group Support Systems: A Descriptive Evaluation of Case and Field Studies. Journal of Management Information Systems 17(3):115-59.

Forgionne G, Gupta J, Mora M (2002) Decision Making Support Systems: Achievements, Challenges and Opportunities. In: Mora M, Forgionne G, Gupta J (eds.) Decision Making Support Systems: Achievements and Challenges for the New Decade. Idea Group, Hershey, PA, pp 392-403.

Genesereth M, Nilsson N (1987) Logical Foundations of Artificial Intelligence. Morgan Kaufmann, San Mateo, CA.

Girle R, Hitchcock D, McBurney P, Verheij B (2003) Decision Support for Practical Reasoning: A Theoretical and Computational Perspective. In: Reed C, Norman TJ (eds.)

Argumentation Machines: New Frontiers in Argument and Computation. Kluwer Academic Publishers, Dordrecht, The Netherlands, pp 55-83.
Gorry GA, Scott Morton M (1971) A Framework for Management Information Systems. Sloan Management Review 13(1):50-70.
Han J, Chang KC-C (2002) Data Mining for Web Intelligence. Computer, November 2002 Issue, 54-60.
Hayes-Roth F (1985) Rule-Based Systems. Communications of the ACM 28(9):921-932.
Holsapple CW, Whinston AB (1996) Decision Support Systems: A Knowledge-Based Approach. West Publishing Company, St. Paul, MN.
Huber G (1984) Issues in the Design of Group Decision Support Systems. MIS Quarterly 8:195-204.
Ignizio J (1990) Introduction to Expert Systems: The Development and Implementation of Rule-Based Expert Systems. McGraw-Hill, New York.
Inmon WH (1996) Building the Data Warehouse. John Wiley and Sons, New York, NY.
Jarke M (1986) Knowledge Sharing and Negotiation Support in Multi-person Decision Support Systems. Decision Support Systems 2:93-102.
Jelassi MT, Foroughi A (1989) Negotiation Support Systems: An Overview of Design Issues and Existing Software. Decision Support Systems 5:167-181.
Karacapilidis N, Papadias D (2001) Computer Supported Argumentation and Collaborative Decision Making: The HERMES system. Information Systems 26(4):259-277.
Karacapilidis N, Pappis C (1997) A framework for Group Decision Support Systems: Combining AI tools and OR techniques. European Journal of Operational Research 103(2):373-388.
Katzav J, Reed CA (2004) On Argumentation Schemes and the Natural Classification of Arguments. Argumentation 18:239-259.
Keen P, Scott Morton M (1978) Decision Support Systems: An Organizational Perspective. Addison-Wesley Publishing, Reading, MA.
Kimball R, Ross M (2002) The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (2nd edition). Wiley, New York.
Kobsa A (2001) Generic User Modeling Systems. User Modeling and User-Adapted Interaction 11(1-2):49-63.
Kobsa A (2002) Personalized Hypermedia and International Privacy. Communications of the ACM 45(5):64-67.
Kolodner J (1993) Case-Based Reasoning. Morgan Kaufmann, San Mateo, CA.
Kreamer KL, King JL (1988) Computer-based systems for cooperative work and group decision making. ACM Computing Surveys 20(2):115-146.
Langley P (1999) User modeling in adaptive interfaces. In: Proceedings of the Seventh International Conference on User Modeling. Banff, Alberta, Springer, pp 357-370.
Maes P (1994) Agents that Reduce Work and Information Overload. Communications of the ACM 37(7):31-40.
McLauglin DI (1995) Strengthening Executive Decision Making. Human Resource Management 34(3):443-461.
Miller JA, Fishwick PA, Taylor SJE, Benjamin P, Szymanski B (2001) Research and Commercial Opportunities in Web-Based Simulation. Simulation Practice and Theory 9:55-72.
Mora M, Forgionne G, Gupta J, Cervantes F, Gelman O (2003) A Framework to Assess Intelligent Decision-Making Support Systems. In: Palade V, Howlett RJ, Jain L (eds.) Knowledge-Based Intelligent Information and Engineering Systems - Proceedings of the 7th Int. Conf. on Knowledge-Based Intelligent Information and Engineering Systems (KES 2003), Oxford, UK. Springer, Berlin, Lecture Notes in AI 2774, pp 59-65.

Nemati H, Steiger D, Iyer L, Herschel R (2002) Knowledge Warehouse: an Architectural Integration of Knowledge Management, Decision Support, Artificial Intelligence and Data Warehousing. Decision Support Systems 33:143-161.

Nonaka I (1994) A Dynamic Theory of Organizational Knowledge Creation. Organization Science 5(1):14-37.

Pearson JM, Shim JP (1995) An Empirical Investigation into DSS Structures and Environments. Decision Support Systems 13:141-158.

Power DJ (1999) Decision Support Systems Glossary. DSS Resources. Available online at http://DSSResources.COM/glossary/ (last accessed: Nov 28, 2004).

Prahalad CK, Hamel G (1990) The Core Competence of the Corporation. Harvard Business Review 68(3):79-91.

Provis C (2004) Negotiation, Persuasion and Argument. Argumentation 18:95-112.

Shim JP, Warkentin M, Courtney JF, Power DJ, Sharda R, Carlsson C (2002) Past, Present and Future of Decision Support Technology. Decision Support Systems 33:111-126.

Sierhuis M, Selvin AM (1996) Towards a Framework for Collaborative Modeling and Simulation. In: Proceedings of the 6th Computer Supported Cooperative Work Conference, November 16-20, Boston, MA. Available online at http://www.compendiuminstitute.org/compendium/papers/SierhuisSelvin-CSCW-1996.pdf (last accessed: Nov 28, 2004).

Simon H (1977) The new Science of Management Decision. Prentice Hall, Englewood Cliffs, NJ.

Smoliar SW (2003) Interaction management: The next (and necessary) step beyond knowledge management. Business Process Management Journal 9(3):337-352.

Spender J (1996) Organizational Knowledge, Learning and Memory: Three Concepts in Search of a Theory. Journal of Organizational Change Management 9(1):63-78.

Sterman JD (2000) Business Dynamics: Systems Thinking and Modeling for a Complex World. McGraw Hill, New York.

Sycara KP (1990) Persuasive Argument in Negotiation. Theory and Decision 28:203-242.

Taylor SJE (2001) Netmeeting: A Tool for Collaborative Simulation Modeling. International Journal of Simulation: Systems, Science and Technology 1(1-2):59-68.

Thomsen E (2002) OLAP Solutions: Building Multidimensional Information Systems (2nd edition). Wiley, New York, NY.

Thompson CA, Göker M, Langley P (2004). A Personalized System for Conversational Recommendations. Journal of Artificial Intelligence Research 21:393-428.

Toulmin SE (1958) The Uses of Argument. Cambridge University Press, Cambridge, UK.

Turban E, Aronson JE (2001) Decision Support Systems and Intelligent Systems (6th edition), Prentice Hall, Upper Saddle River, NJ.

van Eemeren FH, Grootendorst R, Henkemans FS, Blair JA, Johnson RH, Krabbe ECW, Plantin C, Walton DN, Willard CA, Woods J, Zarefsky D (1996) Fundamentals of Argumentation Theory. Lawrence Erlbaum Associates, Mahwah, NJ.

Vennix JAM (1996) Group Model Building: Facilitating Team Learning Using System Dynamics. Wiley, Chichester, UK.

Walton D (1996) Argument Structure: A Pragmatic Theory. University of Toronto Press, Toronto, Canada.

Watson I (1997) Applying Case-Based Reasoning: Techniques for Enterprise Systems. Morgan Kaufmann, San Francisco, CA.

Weiss G (1999) Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. MIT Press, Cambridge, MA,

Wooldridge M (2002) Introduction to Multi-Agent Systems. John Wiley & Sons, New York.

Zuurbier JJ (1992) On the design of group decision support systems. In: Jelassi T, Klein MR, Mayon-White WM (eds.) Decision Support Systems: Experiences and Expectations. IFIP, North-Holland, Amsterdam, The Netherlands, pp 59-69.

21

A Challenging Future for i-DMSS

Nathalie Cassaigne and Lionel Lorimier

The University of Manchester, P.O. Box 88, Manchester, United Kingdom
e-mails: nathalie.cassaigne@manchester.ac.uk, lionel_lorimier@hotmail.com

This chapter reports the findings of exploratory research into the technical and organizational challenges facing i-DMSS for nonoperational decision making and indicates directions for future research. Some challenges arise as a direct result of expert judgments used to compensate for the lack of information inherent in complex, uncertain situations, and the necessity to support intelligently the interaction between decision makers and experts, each of them having different backgrounds, interests, knowledge and cognitive style. Another important challenge results from the manipulation and interpretation of information, especially for expert judgments' aggregation and the providing of advanced functionalities such as forecasting in the context of highly uncertain and complex decisions.

21.1 Introduction

More than ever, the world has become a global society with a global economy. The slightest decision taken on one side of the world might have an unexpected impact somewhere else in the world. Straightforward cause-effect relationships are now less easily found and problem boundaries are more difficult to define. Uncertainty and complexity are becoming common facts leading to the greater recognition of systemic and holistic approaches to problem solving.

Decision makers are confronted by a continuum of situational types. By situation we stress the fact that the environment of a problem should be considered as well as the problem itself. At one end of the continuum, situations can possibly be complicated but not complex, *i. e.* there might be a great number of factors to take into account but their relationships are known or expected, therefore the degree of uncertainty in terms of problem modeling is very low if nonexistent. The frontier of the problem can be defined and formalized relatively easily. At the other end of the continuum, the level of complexity and its underlying uncertainty is such that no solution to the situation can be found. Most decision support systems (DSS) currently available focus on complicated problems. With these problems sorted out, decision makers are now requiring support to tackle more sophisticated problems involving more uncertainty and complexity. These problems are characterized by the fact that emergent properties that may be unexpected and counterintuitive might be a direct or indirect result of the interaction between the various parts or aspects of a situation (Daellenbach 1994).

Another way of looking at this is to consider organizations as decision-making organisms (Mintzberg 1979, Simon *et al.* 1958, Simon 1977) where decisions are categorized, according to an adaptation of the Anthony's model (Anthony 1965), as operational (or day-to-day), tactical and strategic. At one end of a continuum, operational decisions are recurrent; they are based on hard data and follow well-established procedures and rules. At the other end of the continuum, strategic decisions might have a long-term impact on the life of the organization. They are mainly unstructured, infrequent if not totally novel, based on highly aggregated, mostly inaccurate, incomplete, imprecise, unverifiable and rather old external information (Sauter 1997). Strategic decisions are mainly based on knowledge and gut feeling to answer a novel situation, in other words they are characterized by uncertainty and complexity. Tactical decisions are in the middle of the continuum and they have characteristics from both ends.

This chapter focuses on problems that are in the middle of, or close to the end of the most complex problems *i. e.* not quantitatively the most important problems in an organization but the most valuable. In this chapter, they are referred to as nonoperational.

While a number of papers, including (Blair *et al.* 1997, Eom 1999, Eom 2004), have investigated current trends in DSS research, this chapter proposes a different viewpoint by focusing on the challenges brought about by the complexity of the decision to be *intelligently* supported. Therefore, it puts forward the problems that need to be solved to provide a truly intelligent support to nonoperational decision making rather than the type of approach used.

The contributions of this exploratory research will be organized around the decision-making process, in particular, the interrelationships between decision-makers, experts and the underlying model of the decision. Due to the complexity of the nonoperational decisions, it might be necessary for the decision-maker to involve one or more domain(s) experts to identify the possible characteristics of the problem, the decision model, the possible solutions and their impact. Also because of the huge implications on that organization and with regard to other organizations and society at large, the decision might involve the participation (or total involvement) of many decision makers.

The first section of the chapter proposes an overview of current DSS, their major strengths and weaknesses. Section two presents the decision maker, his characteristics in terms of mental model, decision style and his biases. Then the first two tasks the decision maker needs to undertake and their challenges are presented. These tasks are to firstly perform a rough analysis of the situation, then to decide on a panel of experts. Section three discusses the challenges faced by experts in their interaction with the decision maker and between themselves in the case of a heterogeneous panel of experts. Also the aggregation of expert judgments is examined. Section four builds upon the previous sections to examine the decision's model.

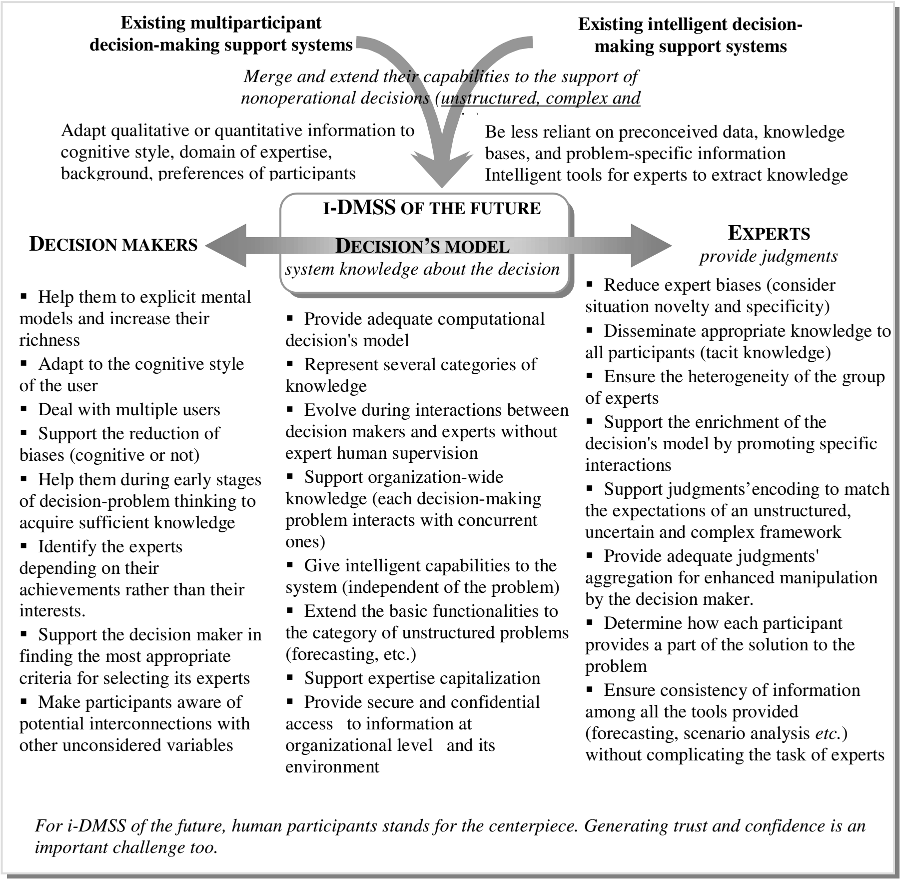


Figure 21.1. Crossrelations between sections and challenges identified in this chapter

Figure 21.1 crossrelates key challenges with the four parts of this chapter: multiparticipant and intelligent-DMSS, the decision maker, the experts and the decision's model.

21.2 Multiparticipant and Intelligent DMSS

As an introduction toward the support, in an intelligent way, of the nonoperational decision-making process, we propose an overview of existing decision-making support systems (DMSS) along two major directions, namely multiparticipant DMSS and intelligent DMSS (i-DMSS). Although the intelligent support of nonoperational decisions cannot be reduced to merely bring closer these two kinds of DMSS, because this would involve taking up many technical and organizational challenges, this approach tailors a better understanding and assessment of the challenges identified in the rest of the chapter.

21.2.1 Multiparticipant Decision-making Support Systems

According to (Holsapple and Whinston 1996) cited by (Chen and Lee 2003), a multiparticipant decision support system consists of four subsystems: (1) a language system that can handle both private and public messages; (2) a problem-processing system that is capable of knowledge acquisition, selection, and derivation; (3) a knowledge system that can store both private and public domain knowledge for multiparticipants; and (4) a presentation system.

More or less elaborated examples comprise organizational DMSS (Carter *et al.* 1992), group DMSS (He *et al.* 2003, Karacapilidis and Pappis 1997, Shih *et al.* 2004) and multiparticipant cognitive DMSS (Chen an Lee 2003). These systems allow a form of support for some unstructured decisions by the use of multiple experts and decision-makers, working together either physically or virtually. The main drawback of these systems is that too often the intelligence is left with the decision maker. For example (Chen and Lee. 2003) provides a cognitive support, enabling information sharing of case-based reasoning but every data is raw therefore it is the role of the decision maker to put things he requires together as the system doesn't offer any aggregation of the huge amount of information made accessible through the system. Other disadvantages (concerning aggregation of judgments, exchanges between experts, *etc.*) are pointed out throughout this chapter with regards to the specific active support of nonoperational decisions. Although multiparticipant DMSS bring advanced communication facilities, they are still failing in supporting effectively these communications. Generally speaking, these systems lack the intelligence necessary to deal with complexity and uncertainty, therefore, leaving that specific function to their users.

21.2.2 Intelligent Decision-making Support Systems

Intelligent-DMSS "would combine the knowledge-based reasoning methods of expert systems with formal methods of decision analysis" (Holtzman 1989) and also with *Artificial Intelligence* (AI) and other computing techniques.

For i-DMSS to be successful, (Holtzman 1989) p.155, states that "*focusing on a well-defined domain is essential for developing meaningful knowledge bases.*" This is possible when dealing with structured to semistructured but not too complex a problem. It is thus not suitable when the objective of these i-DMSS is to support complex situations. Therefore, a challenge for i-DMSS is to be less reliant on preconceived data and knowledge bases.

It is to be noted that AI and other computational techniques (including but not limited to fuzzy logic, neural nets, genetic algorithms, Bayesian networks… and their combinations) (Kasabov and Fedrizzi 1998, Kasabov *et al.* 1999, Toussaint 2002, Tran *et al.* 2004, Watkins *et al.* 1992, Zeleznikow and Nolan 2001) have contributed to improve the degree of intelligence (Mora *et al.*, 2003) conferred to DSS for operational (Eom *et al.* 1998) and tactical (He *et al.* 2003, Kumar *et al.* 1997, Li *et al.* 1997, Matsatsinis and Siskos 2002) decision making. See (Blair *et al.* 1997) for a review of methodologies used to develop such i-DMSS. However, existing i-DMSS are not adapted to fully support complex decision making for several reasons:

- Computational intelligence techniques are model based, even if the model is not always explicitly known in full. Thus, the uncertainty supported is limited to imprecise or vague data. Moreover, for connectionism evolving structures, the training period requires data and experts to be available in order to set up the system, specify the rules (the notion of rules means that an expert interpretation of the data presented to the system provides the conclusion).

- From a computing viewpoint, it is impossible to capture all the facets required by the decision. Too much of the knowledge required about the decision is problem specific, which means that the structure of the model is not known a priori; all required pieces of information are not available and must be compensated for by expert knowledge, that is unfortunately not reproducible.

- Underlying rules have to be understood to interpret the results the system provides. Therefore, the user has to be a domain expert to judge appropriately; that is not often the case in a non-operational decision, where decision makers are often top executives rather than domain experts.

In other words, existing i-DMSS are well adapted to an experts' use, focusing on part of the decision and its context, thus supporting them in retrieving information, extracting knowledge and analyzing possible sub-scenarios (*i. e.* not too complex) for a future use at a time they have to judge and give their opinions. However, they have to take up many challenges, exhibited in this chapter, to deal efficiently with the complexity and uncertainty inherent in nonoperational problems.

21.3 The Decision Maker

The primary objective of decision support tools is to assist the decision maker in performing a specific task. The issue here is not to substitute the system to the decision maker because this can only be considered for structured tasks such as operational tasks for which a large amount of data is available, the objective and factors of the task are well defined and the decision model follows normative rules. The more a task is uncertain and complex, then the more a human being will become necessary.

Therefore, for a system supporting nonoperational tasks to be qualified as intelligent it should take into account - by this we mean assist and support - two types of human being. The decision maker who is the main stakeholder - here many architectures might be used from a sole decision-maker to a group (Marakas 1999) and, in addition, the decision maker might decide to recruit one or many experts to support the decision-making process as described by (Simon 1977) -. In this section, we will focus on the main stakeholder and the many challenges he poses to a DMSS for it to become an i-DMSS. First, we briefly investigate his needs and characteristics through the concepts of mental models and cognitive or decision style and inherent biases. Then, we introduce the first two steps relative to the nonoperational decision-making process because the decision maker plays the major role during these steps.

21.3.1 General Characteristics

21.3.1.1 Mental models

The decision makers who will be mostly confronted with complexity and uncertainty are the top executives who need to make sense of ambiguous information (Mintzberg 1973, Rockart and De Long 1988) cited by (Chen and Lee2003). The human mind cannot cope directly with the complexity of the world because of its limited mental capacity. Therefore, human beings understand the world by constructing in their mind working models of it, which is a way to reduce uncertainty. Top executives use their mental model defined by (Norman 1983) as internal representations of the realities that are developed as a result of interaction with them. They are further defined by (Chen and Lee 2003) as deeply held assumptions and beliefs that enable individuals to make inferences and predictions to simplify their vision of the environment, identifying the problem and its possible solutions by categorizing, classifying their environment in terms of similarities and differences (Mintzberg 1973, Porac and Thomas 1990).

Mental models are often simplified, incomplete and fragmentary if compared with reality or with the system they represent (Johnson-Laird 1983), however they evolve inductively as the user interacts with the system (Jonassen 1995) through perception. As mentioned in (Heuer 1999), perception is a process in which a person constructs his or her own version of reality, a mental model, as a framework for thinking about the reality. Concepts stored in memory (*e. g.* assumptions and tacit knowledge) exert a strong influence on the formation of perceptions from data and information received through sensory systems. In other words, past experience, education, cultural and organizational norms and values, role requirements as well as the details of information affect what information the receiver perceives, how easily he perceives it, and how he processes the received information. According to (Heuer 1999), events and information consistent with expectations originating from experiences, trainings, values, norms, *etc.* are perceived easily, while contradictory ones are distorted or ignored. Mental models are formed rapidly but resist change and therefore condition future perceptions of the reality. Past experience can be counterproductive when a fresh view is required as there is a tendency to assimilate new data to an existing image and, as a result, looking at the same thing from different angles is difficult.

A challenge for i-DMSS is to propose ways to support the decision maker in identifying the mental models he is using (make them explicit instead of tacit) at each step of the decision-making process and increasing their richness.

The objective of any DMSS is/should be to support the decision, not to replace the decision maker whose intuition is good at defining important factors but poor at combining that information (Lu 1995, Lu *et al.* 2001). According to (Zmud 1986), executives mainly need support in problem and opportunity recognition that is

(Mintzberg 1973) recognized as one of their major roles. This chapter identifies other opportunities to support the decision-maker.

21.3.1.2 Cognitive or Decision Styles

The influence of the decision-maker's cognitive style has been the object of numerous studies in the domains of management information systems and knowledge management (Hunt *et al.* 1989, Lu *et al.* 2001, Rowe and Boulgarides 1992, Ruble and Cosier 1990, Zmud 1986).

Rowe and Boulgarides (1992) classifies decision-makers along two continuums. The first one assesses their response to complexity with, on the one hand the need for structure and on the other the tolerance for ambiguity. The second continuum assesses their judgment style or value orientation and defines four cases depending on the one hand, on the decision-makers' approach to problem solving and on the other their approach to tasks and the people concerned.

As a result, (Rowe and Boulgarides 1992) considers four decision types that have been summarized by (Marakas 1999) as the analytical decision maker, who demonstrates a great tolerance for context ambiguity, thrives on challenges, requires great volumes of information, considers a large set of alternatives and enjoys problem solving. However, his orientation toward detail often results in protracted investigations of the problem before any decision is made. The directive decision maker needs structure and shows a low tolerance for ambiguity. He tends to focus on a decision of a technical nature and will use heuristics formalized in the form of policies and procedures. He is motivated by power and status. The conceptual decision maker is described as having a high tolerance to ambiguity and is a "people person", he is achievement oriented and driven by an idealistic emphasis on values and ethics, and he is a creative person trusting his intuition and judgment. Finally, the behavioral decision maker, who, like the directive decision maker, displays a low tolerance to ambiguity that he tries to avoid. He is people oriented rather than task oriented and is motivated by peer acceptance. He only needs a small amount of information to take a decision mainly based on his feelings and instincts.

Turban (1995) advises that a specific process should not be enforced while (Huber 1983) believes that too much has been made of recognizing cognitive styles when implementing DMSS. This chapter claims that the nature of the decision needs to be ascertained before deciding on the usefulness of adapting DMSS to the user's cognitive style. Indeed, whilst on the one hand existing systems mainly deal with operational decisions, therefore, they mainly need to focus on the structure of the task, and as a result, they do not have to take into account the cognitive style of the decision-maker. On the other hand, complex decisions are primarily based on judgment and gut feeling making the decision maker and the experts the key to the success of the decision therefore their decision styles must be taken into account.

A challenge for i-DMSS is to be flexible and adapt to the cognitive style of the user of the system, which can provide an extra challenge when dealing with multiple users such as in multiparticipant DSS. For example, the strength of the directive decision maker confronted with a strategic decision is that this type of decision often requires reacting quickly on a small amount of data and, in these circumstances, intuition might be the only basis for the decision. However, strategic decisions are highly ambiguous and complex therefore they put the directive decision maker under a great deal of stress. Support can be provided to reduce ambiguity by providing access to domain expertise and knowledge (database/person) to help with the identification of the specific problem. Data, information, judgment and knowledge could be grouped into clusters, indicating their source and reliability (in quantitative terms); access can be provided by the system to external rules, policies, procedures, benchmarks developed for similar situations by professional bodies. Whenever possible, findings should be presented in graphical, tabular forms (Cassaigne 2002).

21.3.1.3 Biases

Whatever their cognitive styles and mental model, decision makers will be afflicted to a greater or lesser degree by biases. Bounded by their cognitive limitations, decision makers tend to accept the first solution that “satisfies” their pre-conceived notions instead of looking for the optimal one (Simon 1979); by doing so, they make use of heuristics and analogies.

Using (Tversky ans Kahnerman 1974, 1981), (Turpin and Plooy 2004) has summarized some of the twenty nine cognitive biases identified by the literature on human information processing into three main categories, each corresponding to a type of heuristics. They are the representativeness heuristic, where people believe that a sample or outcome is representative of a bigger class while it is not; the availability heuristic, where easily recollected events appear to have more chance to reoccur than others; the adjustment and anchoring heuristics, where when confronted with a large amount of data, the person selects a particular datum as a starting point (or anchor) and then adjusts that value improperly in order to incorporate the rest of the data. Another possible problem is also identified and called the problem of decision framing. This bias can be interpreted as being related to the use of mental models.

The literature review reported by (Turpin and Plooy 2004) shows that the use of information systems such as decision support systems, executive information systems, *etc.* can sometimes “enlarge” some biases but it appears that there is no clear evidence to support this assumption or to inform it. It says that *“the role of an information system in reducing or increasing task-related biases of information processing is dependent on its design, use and perceived usefulness in relation to the task at hand”*.

When in doubt, it is safer to encourage i-DMSS designers to support the reduction of biases that (Hogarth 1980) identifies as related to the task environment, the complexity of the task, the uncertainty and stress faced by the decision-maker and also his commitment to the task. Therefore, this is another challenge for i-DMSS. According to (Pfeifer and Scheier 1999), p.5, *“all definitions of intelligence have a common denominator related to novelty and adaptivity”*. They mainly refer to the adaptability to a new situation but we claim that the adaptability to the user is also very important because as stated by (Lu 1995, Lu *et al.* 2001) the willingness to use DSS is a function of an individual’s cognitive style, beliefs, and attitudes. A support system that does not interact with the user -in this case, the decision maker - or adapt to him so that it overcome his weaknesses while supporting his strengths has only a limited intelligence.

21.3.2 Rough Analysis Performed by the Decision maker

According to Simon's model (Simon 1979), the decision-making process follows four steps. First, during the intelligence phase, the reality is examined, the problem identified and defined. Then, during the design phase, a model is constructed and validated, and alternatives are developed. Follows a choice phase where the alternatives and the solutions to the model are evaluated. Eventually, the chosen solution is implemented and the success or failure determined. In addition, a feedback mechanism revises each step and provides learning capability.

In a complex situation, the decision maker might require support from analysts and experts. According to (Yiman-Seid and Kobsa 2003) we seek an expert as a source of information, in particular when there is a need for interpretation, or when seeking a consultant.

Following Simon's decision-making process, first a rough analysis is performed by the decision maker with the intention to determine the complexity of the problem and therefore the need to call upon experts. This rough analysis can go as far as identifying the domains involved, correlated problems, and potential scenarios. This analysis will provide the foundations of the decision-maker's model. If he decides he needs support, the decision maker must define the criteria to be used in selecting the most appropriate internal and external experts. He might call upon one and/or the other of two categories of experts:

- The main panel: these experts work on the whole decision-making problem, *i. e.* the decision maker describes the situation to them without simplifying its complex aspects, and it is the role of these experts to develop their own models to understand, then analyze and judge the situation.

- The supporting panel: experts in this group are consulted on demand on specific aspects of the problem and are in possession of a partial or simplified view of the situation.

The decision maker and experts (in agreement with the decision maker for obvious confidentiality issues) have the opportunity of adding new experts to existing groups, depending on the evolution of their understanding of the situation.

21.3.3 Building the Panels of Experts

Extensive research has been developed in the domain of expert finding or expert location. A comprehensive analysis of the current systems and trends is provided by (Yiman-Seid and Kobsa 2003) who highlight that it is part of a broader knowledge management (*ibid.* 2003) and computer-supported cooperative work issues. Based on this chapter, we can conclude that expertise is currently sought using one of three approaches. The first approach builds an expertise profile based on the analysis of the content of explicit sources such as e-mail, documents submitted through the intranet of a particular company. The second approach is based on referral chaining by recommendation from colleagues. The third approach analyses communications such as the exchange of e-mails, participation to discussion groups and search patterns such as World Wide Web browsing patterns. As highlighted by (*ibid.* 2003), most systems attempt to exploit implicit evidence of expertise, as a result, the main shortcoming of the latest approach is that browsing (or even communication in general) merely shows someone's interest not his expertise.

Also (*ibid.* 2003) reports on a number of challenges facing these systems from the need to handle the heterogeneity and distributedness of the information space, to the correct definition of expertise indicators. The concepts used by current systems are a generalization of information-retrieval assumptions and theories that might not be fully adequate for the problem. In addition, according to (*ibid.* 2003), expert-finding is split into two phases: expert identification and selection. The latter is apparently less addressed by researchers than the identification phase. Therefore, a challenge for the i-DMSS is to support the decision maker in finding the most appropriate criteria to select the experts.

The selection of the appropriate panel of experts, depending on the available resources, plays a pivotal role for the future of i-DMSS for nonoperational decisions. Indeed, because of the lack of reliable data, the accuracy of conclusions and forecasts provided by such i-DMSS relies heavily on the quality of the experts used.

21.4 The Expert

As a human being, an expert will be influenced by the same human information-processing mechanism and biases the decision-maker is exposed to, but not to the same extent, depending on their level of expertise. They also build mental models and have a predominant cognitive style. In this section we first propose a model for assessing the level of expertise, then deal with the major facets of the second and third step of the decision process as proposed by Simon (Simon 1979). To this effect, we will explore issues arising during the phases of design and of choice such as interactions between decision maker and experts, exchanges inside a heterogeneous panel of experts, elicitation and aggregation of expert judgments.

21.4.1 Levels of Expertise

The underlying assumption of the Dreyfus model (Dreyfus 1982) (containing six development stages) is that people depend less and less on abstract principles and more and more on concrete experience as they become proficient. In addition, they are less formally aware of monitoring their action and are more and more aware of the context of a situation. The reliance of an expert on his experience depends on his level of expertise. With increased expertise, a person develops the use of reasoning by analogy and the recognition of features depending on the context of the problem. Therefore being able to recognize the level of expertise required to call upon the right expert is important.

Exploratory research based on the authors experience allows us to succinctly describe the ability of the expert to handle the three type of decisions (operational, tactical, and strategic) and what type of support they might need from a i-DMSS in order to use their expertise more effectively. It is to be noted that only the last three levels of expertise as described by Dreyfus were considered because the first three can only cope more or less effectively with operational decisions.

According to (Chernyshenko *et al.* 2003), proficient skill level comes with increased practice that exposes the person to a variety of situations. Aspects appear more or less important depending upon relevance to goal achievement. Contextual identification of similar features and aspects of the task is now possible and memorized principles are used to determine action. There is still a detached commitment relative to deciding, but the proficient person has become involved in the outcome of the decision and understanding of the features and aspects of the task. The proficient person has developed an experienced perspective of the task. However, aspect recognition still needs explicit encouragement not by calling attention to recurrent sets of features but rather by singling out clearly expressed examples. Similarly, it is still necessary to attract his attention toward the recognition of dangerous aspects and the knowledge of guidelines to correct these conditions. This applies to operational decision making where decision support systems can provide such a framework but is vital for tactical decisions. A proficient person might be able to start working on the less-complex tactical tasks indeed, with support they might start being able to tackle the structured part of a tactical decision. The next phase is to become an expert and ultimately a master.

The repertoire of experienced situations for the expert is now vast, so that the occurrence of a specific situation triggers an intuitively appropriate action. The expert is still consciously aware of monitoring his performance and has an involved commitment to all facets of the task. While the master is absorbed and no longer needs to devote constant attention to performance. He can devote his energy to identifying the appropriate perspectives and appropriate alternative solutions.

Both expert and master can deal efficiently and effectively with an operational decision with the master demonstrating more dexterity than the expert does. On the other hand, some nonoperational tasks will still be too complex but they might be able to handle at least partially the unstructured part of the tactical decision even if aspect recognition and recognition of dangerous aspects might still need to be encouraged.

As for strategic tasks, some tasks will remain too complex for any expert whatever their level and domain of expertise. This is because they might have to deal with a totally novel situation that does not show similarity to previous experiences. In this case, the recourse to gut feeling and judgment is the only possibility. Some general skills such as learning how to learn, how to observe, how to single out features might also help approach a novel situation as long as it doesn't overshadow it and therefore become a bias.

An intelligent support system should thus encourage aspect recognition by calling attention to recurrent sets of features and by singling out perspicuous examples. Then, ensure that action is not automatically triggered before recognizing that some features are novel and finally, ensure that expert and master are careful with their use of heuristics and analogies.

21.4.2 Interaction Between Decision Maker and Experts

Ben Haïm, in the preface of his book *Information-Gap Decision Theory* (Ben-Haïm, 2001), introduces the decision making under uncertainty with these words: *"Difficult decision-making, especially under severe uncertainty, is a process of evaluating and revising assumptions, goals, methods, information, preferences etc."* Any confrontation with other participants brings important feedback in order to obtain a better understanding of the complex uncertain decision problem characterized as open-ended, unstructured and a severe lack of information.

Thus, a better understanding of the situation is provided by the interaction between decision maker and experts but also during interactions between experts. Indeed, as stated in (Wolinsky 2002), new knowledge emerges through the mutual interrogation between experts and between experts and decision maker *"by arousing new questions and providing a better understanding of the situation"*. It is a feedback-based process where actors are permanently interacting to improve and refine their understanding of a complex situation in order to get as close as possible to a complete model of the situation. By the end of his rough analysis, the decision-maker might only have gone through a first rough iteration of the intelligence phase or might have also briefly and incompletely considered the design phase of Simon's model of decision-making (Simon 1979). In a normative model, the decision maker would describe the situation to the expert(s) with the least interpretation as possible, but in a descriptive model, we might find that, because of his cognitive style, mental models, information-processing style possibly including an illusion of expertise (Fellner *et al.* 2004), the decision maker might give an account of the situation that reflects his own interpretation.

The i-DMSS should aim at limiting the influence of the decision-maker's personal interpretation of the situation in several ways, for example through cross-related questions where the system identifies potentially involved domains, *etc.* in comparison with past decision-making problems or classical decision-making problems (knowledge base).

21.4.3 Exchanges Inside a Heterogeneous Panel of Experts

As shown more generally for decision processes involving multiple expert participants (Chernyshenko *et al.* 2003, Rowe *et al.* 2004, Stasser *et al.* 1995), sharing and disseminating expert judgments or conclusions to other participants becomes essential to improve the conclusions provided by the panel. Indeed, it acts either as stimuli for new investigation, for instance in the case of conflict, or justifies current thoughts. As these experts may not come from the same domain of expertise, it further requires adapting the presentation of such expertise so that each expert can meaningfully manipulate, appropriate and then use the newly shared or generated expertise. The experts will investigate the situation using their own knowledge, expertise, models, and tools.

Thanks to the advances in computer network technologies and modern telecommunications, virtual teaming (Duarte and Snyder 1999, Lipnack and Stamps 1997, Townsend *et al.* 1998) have taken an increasing importance in organizations, grouping team members with distinct complimentary domains of expertise.

Although they may be geographically distributed, belong to different organizations and even intervene at different levels in the decision process, recent studies (Potter and Balthazard 2002) prove their effectiveness and performance, so long as they interact harmoniously and in a constructive manner, tending even to decrease the effect of some traditional biases.

In addition to the performance of virtual teaming, the computer-supported interaction between experts provides a favorable environment to develop an intelligent support of communications, where the objectives assigned to the panel and its heterogeneousness are taken into account by adapting the information to the cognitive style of the expert and his domain knowledge thus limiting the required efforts. Indeed, different domains of expertise and backgrounds mean different languages and ways of interpreting the same values or judgments. So, it should be fostered by future i-DMSS.

Furthermore, having a heterogeneous group of experts in terms of Dreyfus qualification and in terms of domains might be a bonus when dealing with novel complex situations as it promotes innovative points of view and exploratory ways of thinking but extra research is required to confirm this assumption.

21.4.4 Expert Judgments

Judgment has been defined in Webster's dictionary as arriving at a decision or conclusion on the basis of indications and probabilities when the facts are not clearly ascertained (Webster 2004). Judgment is the key method of coping with uncertainty when the required data is either unavailable or incomplete, ambiguous or contradictory. Judgment is made by combining available information and what the experts bring to the analysis of this information. The strength of judgment depends on the organization of experience and information in the memory of the expert as the process at least includes the ability to remember relevant facts as well as patterns that relate facts to each other and to broader concepts (Heuer 1999).

The judgment encoding for computational manipulation is essential. Indeed, proposing possible judgments (ranking or voting techniques, *etc.*) restricts the expressing power (complexity of the problem is abstracted) and stands for a simplification of the decision's situation that has to be avoided.

In fact, thanks to their analysis, interactions and judgments, the experts and the decision maker, extract and formalize progressively the knowledge (decision model, interpreted information) required to enable the decision maker to assess the potential solutions to the given problem. In the context of nonoperational decision-making problems depicted here, we need to be precise that the experts do not propose any solution as such but only partial views of it for the facets of the decision they have investigated. In addition, they are not in possession of all, if any, of the decision criteria that the decision maker will use to select a solution between the alternatives that are in his possession. It is to be noted that these criteria may be imprecisely expressed and will evolve with the progressive understanding of the problem.

21.4.5 Decision-maker Understanding and Interpretation (or Model)

By definition, the decision maker is not always a domain expert for a given situation but he might need to manipulate or at least to understand the conclusions or solutions proposed by the experts so as to make up his mind and take a decision. Therefore, it is important for the i-DMSS to present information and results in a way that is easy, for a nondomain expert, to comprehend, interpret and manipulate without losing its meaning (Cassaigne 2002, Papamichail and French 2003).

The ideal achievement is that the intelligent support system adapts shared information to the user, whether it is quantitative or qualitative information, so that he can get a valuable and correct understanding of the information, without having the tacit knowledge and models used to analyze and interpret them.

It is an important challenge for future i-DMSS to offer a framework enabling the decision maker to have access to all information and expert judgments emerging from the interaction between experts and between the decision maker and experts leading to the extraction, formalization and dissemination of the appropriate knowledge. By appropriate we mean that this knowledge will support a model of the situation that might not result in an optimal but satisfying decision because in complex decision making, optimization is not achievable. The system should enable the decision-maker to "play" with the information and the judgments provided so that he makes his own viewpoint and elicit even more relevant knowledge, asking for judgment refining, scenarios analysis *etc.* (Wolinsky 2002).

An i-DMSS should compensate biases encountered in the decision process, without adding new ones, and provide intelligent facilities to system users. It is explicitly said in (Chen and Lee 2003) that a decision support system should address the following issues: (1) consciously helping enrich the decision-maker's mental models; (2) facilitating mental model validation and integration; (3) supporting the decision-maker's backward and forward thinking; (4) mitigating judgmental errors due to limited human information-processing capabilities.

21.4.6 Getting and Aggregating Expert Judgments

Experts are consequently in charge of providing their judgments. Each expert covers a part of the decision problem, and an efficient technique for aggregating judgments has to be provided. Therefore, future i-DMSS have to support obtaining the expert judgments required to complete the model of the decision, touching all facets of this complex decision, environment included.

Indeed, no expert has a global view of the problem but brings a piece to the puzzle. Thus, the goal of judgment aggregation is to put together these pieces, given that this puzzle is special since judgments are not disjointed and may be conflicting, in order to build an effective model of decision and provide them to the decision maker. However, the aggregation of expert judgments stands alone for a complex task. Thus, the next subsections investigate the existing ways of aggregating judgments, from a traditional point of view (without any computer support) and then inside the frame of multiparticipant decision support systems.

21.4.6.1 Traditional Aggregation of Expert Opinions (without DSS)

In traditional approaches (Budescu and Rantilla 2000, Budescu *et al.* 2003, Chernyshenko *et al.* 2003, Yaniv 1997), it is the role of the decision-maker (or the judge in a judge/advisor context (Budescu and Rantilla 2000, Sniezek and Van Swol 2001)) to aggregate the advices coming from multiple sources. Decision-makers use various aggregation rules either normative, statistical, heuristic or intuitive, involving consensus finding, averaging, weighting and trimming, *etc.*

Budescu and Rantilla (2000) identifies four distinct categories of factors that affect the aggregation: the decision maker (a person, group or model, its level of experience, its personal expertise, its cognitive style *etc.*), the decision task (its context, its importance, the type and amount of information available, the way information is exposed), the expert advisors (accuracy, credibility, intercorrelations among their forecasts, *etc.*) and the information on which their advice is based (reliability, validity, confidence *etc.*). The i-DMSS should consider these factors when performing an aggregation. In addition, consensus finding on a decision alternative is not the objective since it tends to eliminate originality and points of view in conflict with the majority, whereas they are often located at the cutting edge of innovation and open a wide range of opportunities that are very important in a strategic context. Thus, new directions have to be envisaged to deal with the effective judgments' aggregation.

Undoubtedly, the intelligent computational support of the aggregation process limits the influence of the decision maker on the results of the aggregation. But the system must take the decision maker into account when exposing aggregation results to him. Also, the decision maker should be able to play with the possible aggregation of the judgments so as to possibly let new results emerge.

Moreover, based on their research on judge/advisor systems, (Budescu and Rantilla 2000, Sniezek and Van Swol 2001) affirm that for the aggregation results to be accepted, the aggregation rules should be accurate and induce high levels of confidence. A commonly admitted way of developing confidence in the system is to improve the quality of explanations accompanying the system conclusions (*ibid.* 2001). In addition, in the context of a group of experts working together for a decision-maker (or by extension a group of decision-makers), two main problems can be identified:

- First, they tend to oversimplify the problem by investigating only parts of the decision aspects: in the case of a group DSS, the definition of alternatives is highly correlated with the appropriate selection of participants as no appropriate support is given to avoid the biases resulting from a group interaction. For an example of these biases see (Schulz-Hardt *et al.* 2002).

- Secondly, in the case of a process where several participants of an organization intervene at different steps, the solution proposed by the existing systems is to categorize the information and abstract some links as part of the aggregation. Unfortunately, the solutions adopted by group DSS to synthesize the judgments of the participants for the decision maker does not give the decision maker the possibility of balancing the judgments depending on what he really wants to achieve.

21.4.6.2 Aggregation in the Case of GDSS

Although neither the group decision support systems (GDSS) nor the judge/advisor systems - which do not authorize explicitly a communication between the advisors contrary to GDSS - offer an adapted support for nonoperational decision making, the investigation of GDSS contributes to exhibit some of the challenges involved in providing an effective aggregation of the experts' judgments.

In a group decision-making framework, the aggregation of the judgments of participants is crucial, since they have some decision power and a decision has to emerge from the group interactions. The aggregation has been traditionally solved by the use of techniques based on either ranking or voting strategies (Puuronen *et al.* 1998) such as the Delphi Method (Linstone and Twoff 1975), with an underlying intended goal of reaching a consensus. Several GDSS have integrated the advantages of AI techniques to offer a computer-based support of the process, for instance by the use of fuzziness or statistical approaches to aggregate individual preferences expressed on decision alternatives despite an imprecise qualitative scoring of these alternatives (He *et al.* 2003, Karacapilidis and Pappis 1997, Shih *et al.* 2004).

Rowe *et al.* investigate in (2004) the judgments changes during Delphi-like procedures and shows that feedback of results to the group after each step of the Delphi procedure is required to get closer to the good solution but, on the other hand, they warn against the format adopted for the feedback. Thet also put forward the problem of the appropriate panelists' selection to get effective results, and the bias of majority influence that do not foster original points of view and uncommon judgments.

Finally, to bring together the virtual teaming solution and the requirements of the nonoperational decision-making process, i-DMSS of the future will have to counterbalance the interests of expert consultants as suggested by (Wolinsky 2002), and thus ensure the veracity of the information shared. Indeed, the involvement of internal experts from different services and organizational levels and possibly distinct domains of expertise, contributes to group people concerned by the possible decision and its impact, and therefore with potentially incompatible personal interests.

In the light of this short review, it is a very important challenge for i-DMSS to find a new way to aggregate judgments depending on the decision problem and the wish of the decision maker. Furthermore, if many tools are provided by the system to the decision-maker (long-term forecasting, scenario analysis, trends, *etc.*), the i-DMSS will have to ensure consistency between information provided by all these tools without complicating the task of the experts that participate in the decision-problem analysis.

21.5 The Decision Model

The decision model regroups in fact the knowledge stored in the system about the situation and is permanently evolving towards enrichment during the decision-making process through interactions between decision maker and experts. As the system creates the bridge between experts and decision makers, the decision model links them to the specifically addressed nonoperational problem.

21.5.1 Knowledge in the System

The decision model represents a part of the knowledge available for i-DMSS to actively support the decision-making process. Several categories of knowledge are cohabiting, and it is a major challenge for future i-DMSS to propose an adequate framework, sufficiently flexible to evolve without requesting expert human supervision. This section considers first the knowledge aspect then the capitalization of expertise at the organizational level.

21.5.1.1 Organization-wide Knowledge

Obviously, the decisions taken within an organization have an impact on one another involving some necessary trade offs and possible conflicts. In addition, the structure of the organization (Mintzberg 1979) and the complexity of the task have an impact on the decision. Therefore, they should be considered as part of a portfolio of decisions, by providing an overall and possibly integrated model of the decisions. Thus, a forecasting functionality that examines the potential impact of a decision on the organization has to take into account other decisions, in particular those that may be made during the time required to perform the decision analysis, with the risk of providing invalid forecasts if a conflicting decision is made.

21.5.1.2 Intelligent Capabilities

As the system has to evolve and capture knowledge without a global expert supervisor, it is essential to transfer to it adequate intelligence - by this we mean not problem specific - also context-specific interpretation and judgments need to be formally captured so as to learn from experience once the solution has been implemented. The aim is to support the experts and decision makers, during their investigation of a specific problem, so that the system acquires, into its decision model, the knowledge required for the active intelligent support of the nonoperational analyzed decision problem.

21.5.1.3 Toward a Computational Representation of the Decision Model

As for each attempt to computerize knowledge into a computational model, many researches will have to be led in order to develop an appropriate decision model.

Indeed, due to the impossibility of modeling all aspects of the decision into a computational decision model, it requires acceptance of a severe degree of uncertainty. Thus, the complexity and uncertainty makes problems harder to solve as stated throughout this chapter. For instance, forecasting the future in an uncertain complex situation is a major challenge. Nevertheless, an observation made by researchers in strategic decision making (Einhorn and Hogarth 1987) justifies the consideration of a forecasting functionality. Indeed, they state that executives, when facing strategic decisions, are involved in two kinds of thinking: looking backward to understand the past and looking forward to predict the future. Furthermore, forecasting the impact of a nonoperational decision requires assessing its impact on the organization (at several levels) and on its environment. Therefore, a number of avenues can be explored such as a dynamical-model forecast or a traditional probability-based modeling, and the only solution is that experts build up progressively the full decision-specific model with the support of i-DMSS to provide forecast ability.

The positive aspect of uncertainty for the organization is that every competitor is in the same situation of under-information, and uncertainty becomes correlated to risk and opportunity. The degree of confidence is proposed by (Ben-Haïm 2001) to deal with uncertainty as a potentiality for new opportunities.

21.5.2 Expertise Capitalization at Organizational Level

The i-DMSS should support decision-makers and experts during their exchanges with an intelligent system and should also use this opportunity to capture the expertise (explicit and implicit (Nonaka and Takeuchi 1995)) in order to reuse it in the future. Indeed, the system has to endeavor to acquire and capitalize on the expertise used during the process. Although it is a utopian idea to substitute computing experts for human participants, some of their expertise, such as their interpretation of tools for instance can be integrated into the system. The more facilities an i-DMSS provides for tool and model creation by experts and decision makers, the more expertise it is possible to capitalize. The system should also offer access to every source of internal and external information required by the experts and the decision maker. It is an organizational challenge, with possible security and confidentiality issues, to enable the wider possible access to information.

21.6 Conclusion

Despite the extensive advances made by decision-making support systems and the variety of fields they cover, an effective intelligent support of the decision-making process depicted in this chapter remains at stake. The unstructured nature of the non-operational decisions plays an important part in this relative lack of active support. Indeed traditional (intelligent)-DSS have concentrated their efforts on operational and low-level tactical decision problems, known for providing good conditions to obtain a computational model of the decision, even if they often require the use of experts to get it (Cassaigne and Singh 2001).

The problems addressed in this chapter are, by nature, unique and it would be a mistake to consider them as fully identical although they might look the same. As the image introduced by the well-known *Butterfly Effect*, the complexity of the decision addressed requires particular attention when attempting to develop a model of the decision. Indeed, a tiny difference between the situations, not perceptible at the human level and thus not captured into the decision model, could lead to the opposite result when using the built DMSS for a future situation that seems similar.

For organizations and decision makers, nonoperational decision-making problems are challenging because of their degree of complexity and stress. And also because they are novel situations to which the decision maker and to a certain extent the experts have not previously been confronted with and for which few, if any, data and pieces of information are available. Therefore, it is not surprising that they offer a very challenging but also promising direction for future research in i-DMSS as identified in this chapter.

Although achieving a system that would propose an answer to all the challenges identified in this chapter could be difficult, it is important for i-DMSS dealing with nonoperational problems to give a privileged attention to the human beings participating in the decision-making process because they represent the cornerstone of the system.

I-DMSS should aim at decreasing the influence of biases, facilitate the interaction between decision makers and experts and support the aggregation of judgment without further simplifying the problem. It should also support the unique perspective of the decision maker providing him with ways to interpret expert judgment and be in explicit control of his own decision model. Obviously, the notion of trust in the system is even more crucial with a potentially harmful strategic decision because it is difficult, if not impossible, to set up a comparative experiment to demonstrate the effectiveness of the system against nonsupported decisionmaking.

References

Anthony R (1965) Planning and control systems. Harvard Graduate School of Business Administration.

Ben-Haïm Y (2001) Information gap decision theory: decisions under severe uncertainty. Academic Press.

Blair A, Debenham J, Edwards J (1997) A comparative study of methodologies for designing IDSSs. European Journal of Operational Research. 103(2): 277-295.

Budescu DV, Rantilla AK (2000) Confidence in aggregation of expert opinions. Acta Psychologica. 104(3): 371-398.

Budescu DV, Rantilla AK, Yu H-T, Karelitz TM (2003) The effects of asymmetry among advisors on the aggregation of their opinions. Organizational Behavior and Human Decision Processes. 90(1): 178-194.

Carter GM, Murray M, Walker RG, Walker WE (1992) Building Organizational Decision Support Systems. Boston, MA: Academic Press.

Cassaigne N, Singh MG (2001) Intelligent decision support for the pricing of products and services in competitive consumer markets. IEEE Transactions on Systems, Man and Cybernetics, Part C. 31(1): 96-106.

Cassaigne N (2002) The Dashboard: A knowledge conversion tool. Proceedings of the IEEE International Engineering Management Conference

Chen JQ, Lee SM (2003) An exploratory cognitive DSS for strategic decision-making. Decision Support Systems. 36: 147-160.

Chernyshenko OS, Miner AG, Baumann MR, Sniezek JA (2003) The impact of information distribution, ownership, and discussion on group member judgment: the differential cue weighting model. Organizational Behavior and Human Decision Processes. 91(1): 12-25.

Daellenbach HG (1994) Systems and Decision Making: a Management Science Approach. Ltd JWS.

Dreyfus SE (1982) Formal models versus human situational understanding: inherent limitations in the modeling of business expertise. Office: Technology and People. 1: 133-155.

Duarte DL, Snyder NT (1999) Mastering Virtual Teams. San Francisco: Jossey-Bass.

Einhorn HJ, Hogarth RM (1987) Decision making: going forward in reverse. Harvard Business Review. 65: 66-70.

Eom SB, Lee SM, Kim EB, Somarajan C (1998) A survey of Decision Support System applications (1988-1994). Journal of the Operational Research Society. 49(2): 109-120.

Eom SB (1999) Decision support systems research: current state and trends. Industrial Management Data Systems. 5: 213-221.

Eom SB (2004) The changing structure of decision support systems research: an empirical investigation through author cocitation mapping. Proceedings of the 2004 IFIP International Conference on Decision Support Systems (DSS2004): Decision Support in an Uncertain World. Prato, Tuscany

Fellner G, Guth W, Maciejovsky B (2004) Illusion of expertise in portfolio decisions: an experimental approach. Journal of Economic Behavior & Organization. 55: 355-376.

He S, Song R, Zhao Q (2003) Design of uncertain group decision support system and its application in intelligent transportation management. Proceedings of IEEE Conference on Intelligent Transportation Systems. 1724-1729.

Heuer RJ (1999) Psychology of Intelligence Analysis. Center for the Study of Intelligence Central Intelligence Agency, USA.

Hogarth RM (1980) Judgement and choice: the psychology of decision. New York: John Wiley & Sons.

Holsapple CW, Whinston AB (1996) Decision support systems: a knowledge-based approach. West Publishing.

Holtzman S (1989) Intelligent Decision Systems. Addison-Wesley, USA.

Huber GP (1983) Cognitive style as a basis for MIS and DSS designs: Much ado about nothing? Management Science. 29(5): 567-597.

Hunt RG, Krystofiak FJ, Meindl JR, Yousry AM (1989) Cognitive style and decision making. Organisational Behavior and Human Decision Processes. 44: 436-453.

Johnson-Laird PN (1983) Mental models (6th printing 1995). USA: Harvard University Press.

Jonassen DH (1995) Operationalizing mental models: strategies for assessing mental models to support meaningful learning and design-supportive learning environments. Proceedings of CSCL '95. Indiana, USA

Karacapilidis NI, Pappis CP (1997) A framework for group decision support systems: combining AI tools and OR techniques. European Journal of Operational Research. 103(2): 373-388.

Kasabov N, Fedrizzi M (1998) Fuzzy neural networks and evolving connectionist systems for intelligent decision support. Proceedings of IFSA'98

Kasabov N, Deng D, Erzegovezi L, Fedrizzi M, Beber A (1999) Hybrid intelligent decision support systems and applications for risk analysis and discovery of evolving economic clusters in Europe. AIT'99.

Kumar N, Krovi R, Rajagopalan B (1997) Financial decision support with hybrid genetic and neural based modeling tools. European Journal of Operational Research. 103(2): 339-349.

Li X, Ang CL, Gay R (1997) An intelligent scenario generator for strategic business planning. Computers in Industry. 34(3): 261-269.

Linstone HA, Turoff M (1975) The Delphi method techniques and applications. Addison-Wesley Publishing Company.

Lipnack J, Stamps J (1997) Virtual teams: reaching across space, time, and organizations with technology. New York: John Wiley & Sons.
Lu H-P (1995) Managerial behaviours over MIS growth stages. Management Decision. 33(7): 40-46.
Lu H-P, Yu H-J, Lu SSK (2001) The effects of cognitive style and model type on DSS acceptance: an empirical study. European Journal of Operational Research. 131: 649-663.
Marakas GM (1999) Decision Support Systems in the 21st century. Prentice-Hall.
Matsatsinis NF, Siskos Y (2002) Intelligent support systems for marketing decisions. International series in operations research and management science.
Mintzberg H (1973) The nature of managerial work. Harper & Row, New York.
Mintzberg H (1979) The structuring of organizations. Prentice-Hall.
Mora M, Forgionne G, Gupta J, Cervantes F, Gelman O (2003) A framework to assess intelligent decision-making support systems, in LNCS: Knowledge-Based Intelligent Information and Engineering Systems: 7th International Conference, KES 2003 Oxford, UK, September 3-5, 2003. Springer-Verlag. 59-65.
Nonaka I, Takeuchi H (1995) The Knowledge Creating Company. Oxford University Press.
Norman DA (1983) Some observations on mental models, in Mental Models. Stevens A, (Ed.). Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers. 7-14.
Papamichail KN, French S (2003) Explaining and justifying the advice of a decision support system: a natural language generation approach. Expert Systems Applications. 24(1): 35-48.
Pfeifer R, Scheier C (1999) Understanding Intelligence. The MIT Press, USA.
Porac JF, Thomas H (1990) Taxonomic mental models in competitor definition. Academy of Management Review. 15(2): 224-240.
Potter RE, Balthazard PA (2002) Virtual team interaction styles: assessment and effects. International Journal on Human-Computer Studies. 56: 423–443.
Puuronen S, Terziyan V, Omelayenko B (1998) Experimental investigation of two rank refinement strategies for voting with multiple experts. Artificial Intelligence, Donetsk Institute of Artificial Intelligence. 2: 25-41.
Rockart JF, De Long DW (1988) Executive Support Systems. Dow Jones Irwin, Homewood, IL.
Rowe A, Boulgarides JD (1992) Managerial decision-making: a guide to successful business decision. Macmillan Publishing Company.
Rowe G, Wright G, McColl A (2004) Judgment change during Delphi-like procedures: the role of majority influence, expertise, and confidence. Technological Forecasting and Social Change.
Ruble R, Cosier R (1990) Effects of cognitive styles and decision setting. Organisational Behavior and Human Decision Processes. 46: 283-295.
Sauter VL (1997) Decision Support Systems: an applied managerial approach. John Wiley and Sons Ltd.
Schulz-Hardt S, Jochims M, Freya D (2002) Productive conflict in group decision making: genuine and contrived dissent as strategies to counteract biased information seeking. Organizational Behavior and Human Decision Processes. 88: 563–586.
Shih H-S, Wang C-H, Lee ES (2004) A multiattribute GDSS for aiding problem-solving. Mathematical and Computer Modelling. 39: 1397-1412.
Simon HA, March JG (1958) Organizations. John Wiley & Sons.
Simon HA (1977) The new science of management decision. Prentice-Hall, 3rd edition.
Simon HA (1979) Rational Decision Making in Business Organizations. American Economic Review. 69(4): 493-513.
Sniezek JA, Van Swol LW (2001) Trust, confidence and expertise in a Judge-Advisor System. Organizational Behavior and Human Decision Processes. 84(2): 288-307.

Stasser G, Stewart DD, Wittenbaum GM (1995) Expert roles and information exchange during discussion: The importance of knowing who knows what. Journal of Experimental Social Psychology. 31: 244-265.
Toussaint M (2002) A neural model for multi-expert architectures. Proceedings of the International Joint Conference on Neural Networks (IJCNN 2002)
Townsend A, DeMarie S, Hendrickson A (1998) Virtual teams: technology and the workplace of the future. Academy of Management Executive. 12: 17–29.
Tran C, Abraham A, Jain L (2004) Decision support systems using intelligent paradigms. International Journal of American Romanian Academy of Arts and Sciences.
Turban E (1995) Decision support systems and expert systems. Prentice Hall, Englewood Cliffs, NJ.
Turpin M, Plooy Nd (2004) Decision-making biases and information systems. Proceedings of the 2004 IFIP International Conference on Decision Support Systems (DSS2004): Decision Support in an Uncertain World. Prato, Tuscany
Tversky A, Kahnerman D (1974) Judgement under uncertainty: heuristics and biases. Science. 185.
Tversky A, Kahnerman D (1981) The framing of decisions and the psychology of choice. Science. 211(453-458).
Watkins PR, Lin TW, O'Leary DE (1992) AI integration for enhanced decision support. Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences. 133-144.
Merriam-Webster Online Dictionary © 2004 Merriam-Webster, Incorporated
Wolinsky A (2002) Eliciting information from multiple experts. Games and Economic Behavior. 41: 141–160.
Yaniv I (1997) Weighting and trimming: heuristics for aggregating judgments under uncertainty. Organizational Behavior and Human Decision Processes. 69(3): 237-249.
Yiman-Seid D, Kobsa A (2003) Expert-finding systems for organisations: problem and domain analysis and the DEMOIR approach. Journal of Organizational Computing and Electronic Commerce. 13(1): 1-24.
Zeleznikow J, Nolan JR (2001) Using soft computing to build real world intelligent decision support systems in uncertain domains. Decision Support Systems. 31(2): 263-285.
Zmud RW (1986) Supporting senior executive through decision support technologies: a review and directions for future research, in Decision Support Systems: A Decade in Perspective. Sol HG, (Ed). Elsevier Science, Amsterdam.

22

A Software Laboratory for Advancing Decision Support Simulation

Gerald R. Wagner[1], David Paradice[2] and James Courtney[3]

[1] Peter Kiewit Institute, University of Nebraska – Omaha
[2] College of Business, Florida State University
[3] College of Business, University of Central Florida

This chapter describes a new software product called the Planners Lab©. The software is a DSS—in this case an abbreviation for *Decision Support Simulator* instead of the traditional *Decision Support Systems*. The software is intended to be used as a laboratory for academic teaching, research and consulting. Consequently, there is no purchase cost for academic institutions to have access to the software on their computers. The chapter describes the current state of the software, giving inferences to how this software can improve decision making. Several potential research streams are described as well. The chapter ends with a discussion about the new breed of DSS developer.

22.1 Introduction

This chapter describes a new software product called the Planners Lab©. The software is a DSS—in this case, an abbreviation for "*Decision Support Simulator*" instead of the traditional "Decision Support Systems." The software is to be used as a laboratory for teaching, research and consulting at several academic institutions. To help realize this goal, there is no purchase cost for academic institutions to acquire the software for access on their computers. Any course that includes modeling, simulation, quantitative analysis, DSS, financial analysis and planning or engineering economics is a candidate for using the software.

The Planners Lab is one of the first business decision support applications to use a development platform that supports real-time and animated interaction (Macromedia FlashMX™). This includes the ability for decision makers to "play" with assumptions to reflect alternative views of the future with alternative scenarios. The software is a starting point to achieve the following vision: Business software should provide decision makers with engaging experiences, as are currently available in computer gaming. Just as gamers receive an immediate visual response to their actions, decision makers should have comparable feedback experiences with software-based simulators.

Visualization technologies for interactive graphics, 3D visualization, maps, video, audio and animation have greatly influenced entertainment, training and presentation of historical data from databases. However, such technologies have made almost no inroads into the domain of real-time business decision making. This software brings decision support up to date with current technologies. In doing so, it provides a platform that conceives of decision-making processes in ways never before supported with computer technologies.

Every DSS is an assemblage of assumptions about the future. Assumptions may come from databases of historical performance, market research and decision maker minds, to name a few. Most assumptions about the future come from accumulated experiences in decision-maker minds in the form of opinions. In examining assumptions, we begin to focus on why a particular formula or variable value exists in a model. This focus is very different from focusing on why a variable changes in a particular way (Kosy and Wise 1983, Paradice and Courtney 1987). Explanations of variable changes focus on the past; the Planners Lab focuses on the future.

Different scenarios emerge from different sets of assumptions. The credibility of a scenario—and its impact on decision making—may critically depend on the ability of decision makers to have "hands-on" control and witness the consequences of various simulated decision sequences.

This chapter briefly describes the technological foundation on which the Planners Lab is based. It describes a related software package and its compatibility with the Planners Lab. The current implementation of the Planners Lab is described in the following section, followed by several examples for research involving decision support simulators. The chapter concludes with descriptions of the software developers of the future and a brief summary.

22.2 Technological Foundation and Related Work

There is a need for software that allows business managers to review and manipulate the assumptions that underlie a decision-making scenario. This is what the Planners Lab software aims to do. A product with a similar philosophy existed in the 1980s, and was founded, developed and commercialized by Dr. Gerald Wagner. The product was called IFPS™, or *Interactive Financial Planning System*; it was developed and supported by Execucom, located in Austin, Texas. IFPS was used by several hundred academic institutions, a goal currently held by the developers of the Planners Lab.

22.2.1 Decision Support Simulators (DSS) to Rehearse the Future

Decision Support Simulators need several characteristics that can be mapped directly into the decision making process. In the early stages of a decision-making process, they should: (1) support examination and discussion of stakeholders' assumptions; and (2) provide a training and collaboration environment for individuals, groups and organizations to build their understanding of the assumptions in complex operations.

In the latter stages of a decision process, DSS should: (1) provide the basis for decision makers to anticipate how future results would be affected by alternative assumptions; (2) help assess the outcome of alternative interventions and strategies; (3) present results visually to provide rapid and clear understanding and communication of assumptions among decision makers; and (4) enable teams in different locations and with different responsibilities to take part in rehearsals prior to committing to a decision.

Representation of future scenarios must be made meaningful to nonexperts participating in the decision process, must accurately represent the "real-world" environment the simulation addresses, and must leverage domain experts' knowledge by providing displays and reports at multiple levels of detail. Each decision participant should be able to recognize the situation at hand and feel empowered to respond. Usability is based on a foundation of simulators with understandable and believable assumptions combined with interactive visualization.

The authors wish to recognize Dr. Peter G. W. Keen for his term "*rehearsing the future*" as the purpose of DSS. This will be the topic of one of his forthcoming books.

22.2.2 Advanced Visualization Technologies to Witness Results from Decision Support Simulators

Visualization technology can enable decision makers to immediately witness the consequences of changed assumptions. While a deep and broad base of visualization technology, research and practice exists, most of the past and current work deals with visualization of objects, rather than simulated scenarios to support decision makers.

Representative areas utilizing extensive visualization include: (1) mining large quantities of data in search of patterns to reveal information about hidden trends or associations; (2) computer gaming that provides experience-rich and responsive, real-time environments; (3) advanced cartography and remote sensing that employ dynamic maps and image time series to convey spatio temporal information; (4) medical imaging technologies; and (5) 3D modeling used in architectural and construction engineering to assess alternative structure designs.

Figure 22.1 is an example of interactive 3D visualization. This is from research at George Mason University.

22.2.3 Macromedia Flash

Flash™ is a software development tool created by Macromedia, Inc. It is most well known for creating Web sites with animation, interactive menus and sound.

Flash began its life as an animation tool specifically designed to create small, efficient animations by using vector lines as opposed to bitmap images. This allows Flash movies to be smaller than other types of animation found on the Web, such as animated GIFs. With a Flash movie, animations can be longer and much more complex while still retaining a relatively small file size and, in turn, a quick download time.

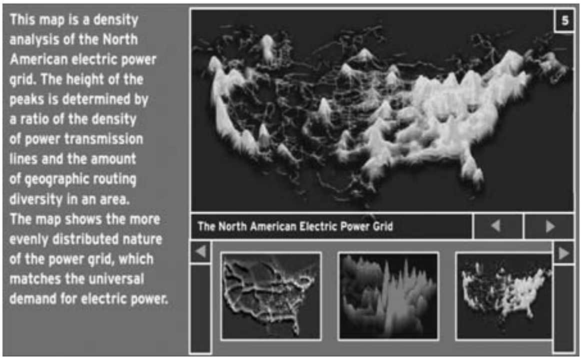


Figure 22.1. Interactive 3D visualization from George Mason University

What has made Flash better than animated GIFs is interactivity—buttons and alternate paths based on what the user clicks on in a Flash movie. Over the years, this ability to assign interaction to elements in Flash has matured into its own robust scripting language called ActionScript.

Today, Flash has grown from its humble beginnings as a Web animation tool into a full-fledged rapid application development platform. It gives developers the ability to create anything from banner ads and animations to entire Web sites—even full-length cartoons for television and film.

The latest version of Flash is especially powerful at creating rich Internet applications and occasionally connected applications that can run on mobile phones and PDAs as well as standard desktop computers. These applications take advantage of the user's Internet connectivity when available and can cache data for use when connectivity is not present.

These tools are made even more powerful by Flash's ability to handle data in XML format and connect to many common data sources and Web services. Because Flash movies are small, these applications are quickly downloadable from the Web or easily sent to others via e-mail.

All these features help Flash developers to create application interfaces that are much more compelling than most software found today. Flash interfaces often have an almost indescribable level of delightfulness to them—the "wow" factor –that keeps users focused and entertained.

Although today's application interfaces are still mostly static, Flash-based applications are defining a future where sound, animation and aesthetics can be effectively combined to create software that is appealing and delightful.

22.2.4 Xcelsius

Xcelsius data-presentation software utilizes the features of Macromedia Flash™ to give users the ability to create visually compelling and interactive charts.

Utilizing an intuitive and easy-to-use, point-and-click interface, Xcelsius converts ordinary Excel spreadsheets into dynamic Flash-based presentations in three steps: by importing an existing Excel spreadsheet; creating interactive data presentations; and outputting the presentations to PowerPoint, Outlook or the Web.

Next, Xcelsius enables the presentation viewer to interact with the data and perform real-time, what-if analyses using animated charts, dials and sliders. For real-time, enterprise-wide dashboards, Xcelsius provides connectivity to company databases via XML and Web services. Xcelsius data presentations are self-contained Flash files that can be deployed to a Web site, portal or PowerPoint presentations—all the while maintaining connectivity to their defined XML data sources.

Xcelsius is tightly integrated with Microsoft Office. Interactive Xcelsius data-presentations can be exported to PowerPoint or Outlook. They can run on any PC, Mac, Tablet PC, handheld or other device with the Macromedia Flash player. Xcelsius is a proprietary software product from Infommersion, Inc.

22.2.5 The Planners Lab

The Planners Lab is built using Microsoft C#.NET and Macromedia Flash MX. The primary tasks of C# include the authentication of user login, handling the flow of model data to and from the database, validating and parsing the model, solving the model, generating Excel files, saving the data to the SQL database, and communicating with FlashMX.

The data flow between Flash MX and C# is handled using XML. The communication with the database is established using SQLOLEDB data provider. C# also generates Excel files that contain the valid equations provided in the model. This is accomplished using the Primary Interop Assemblies of Microsoft Office and the COM support features of .NET technology.

An overview of the Planners Lab software is shown in Figure 22.2. The requirements for using the application are Microsoft Windows 98 or higher, Microsoft SQL Server, Microsoft Office 98 or higher, Microsoft .NET Framework, and 1024 by 768 resolution or higher.

22.3 The Planners Lab Software

The Planners Lab is not created solely for financial planning, but that is its initial primary focus. Successful financial planning is a core critical success factor for every organization. Today, the *de facto* standard for financial analysis software is MS Excel©, even though its use is often fraught with errors. This claim has been extensively documented (Panko 1998). One reason for the errors is an inability to read and comprehend the assumptions in complex computations. This is an issue directly addressed by the Planners Lab.

22.3.1 The Planners Lab Software – Overview

In order to overcome a major source of errors inherent in widely used spreadsheet software, the Planners Lab is aimed at letting decision makers describe their financial plans in their own words and with their own assumptions. The product's *raison d'etre* is that a *simulator should facilitate a conversation with the decision maker in the process of describing business assumptions*. All assumptions are described in English equations (or the user's native language).

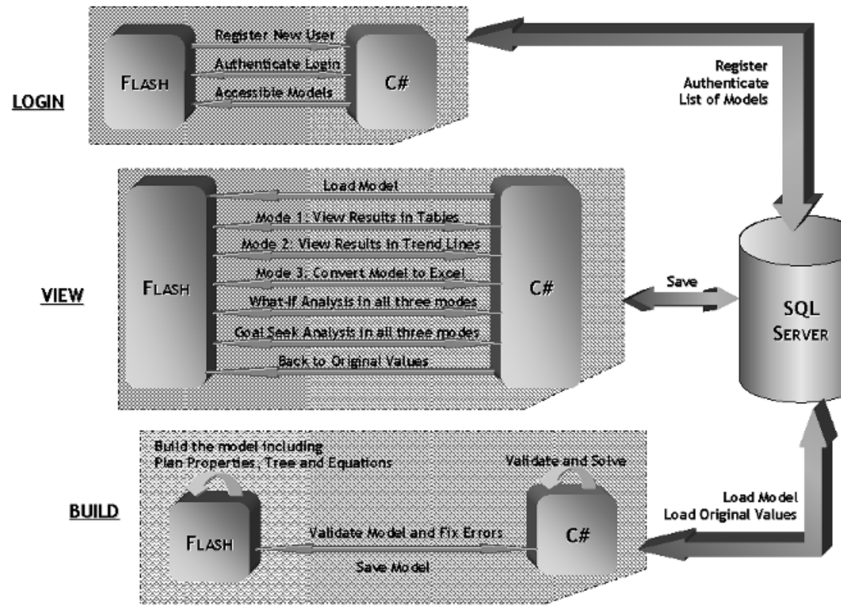


Figure 22.2. Overview of Planners Lab architecture

As can be seen in the simple example below, the Planners Lab is a planning or modeling language and not an Excel-like calculator. Excel is a calculator performing arithmetical operations based upon a matrix of columns and rows. Calculations for a cell are viewed one cell at a time; with the Planners Lab all assumptions for all variables are readable in one view. The following are example assumptions in English using five time periods:

COLUMNS 2005, 2006, 2007, 2008, 2009
Price = 15, PREVIOUS * 1.05
Volume = 2000, 3000, 4000, 5000, 6000
Revenue = Price * **Volume**

COLUMNS indicate a five-column plan where the column titles are 2005 through 2009. Price is 15 in the year 2005. For each subsequent year, the price is the previous year's value is multiplied by 1.05 or an increase of 5 percent each year over the previous year; Volume is 2000 in the year 2005 and increases to 6000 in the year 2006; revenue is price multiplied by volume. For those familiar with MS Excel, the difference between a calculator (MS Excel) and a planning language (Planners Lab) is apparent.

When solved, the simple tabular report would be as shown below.

	2005	2006	2007	2008	2009
Price	$15.00	$15.75	$16.54	$17.36	$18.23
Volume	2000	3000	4000	5000	6000
Revenue	$30 000	$47 250	$66 150	$86 822	$109 400

Every model, regardless of size or complexity, is conceptually no more difficult to read than the above simple example. With this brief introduction, a manager could read and understand a considerably more complex model. The software's hierarchical model structure will accommodate any plan configuration for any organization, such as a multinational company strategic plan with several profit centers, a budget for a huge health care operation with several cost centers, or a financial plan for an old town boutique. The overall plan is structured into small self-contained chunks that are easy to understand and manage. Every plan or model is a custom development and not someone else's interpretation of the user's needs. Decision makers are not forced into someone else's black box.

A Planners Lab model can be saved as an Excel spreadsheet by clicking on the Save as Excel button. Thus, the user can create the model logic in understandable English equations and then access the Excel spreadsheet for reporting, charting, exporting to other systems, and so forth.

The collection of equations is a model *that tells a story for a particular scenario*. Example applications include strategic planning, financial planning, budgeting, project investment analysis, merger and acquisition analysis and sales forecasting.

22.3.2 The Planners Lab Software – Detail

The model-building structure is based upon the philosophy of building large-scale models by creating manageable and understandable small chunks in some hierarchical structure containing nodes. Hierarchical structures are widely used and easily comprehended. The most common example of a hierarchical structure is the organizational chart. The nodes in the hierarchy contain equations that can be entered in any order and, except for reserved words, are entirely in the words chosen by the user.

The example shown in Figure 22.3 shows a model structure with thirteen nodes. The Corporate Summary node has been clicked that displays the equations for that node. These thirteen nodes are all at the same level in the hierarchy but nodes can have any level of subnodes. Any number of nodes can be opened at a time and equations displayed.

Variable names on the left side of the equal signs are shown in red. Reserved words or language words are shown in blue and must be in all capital letters. Reserved words have specific meaning. Examples of reserved words are PREVIOUS, SUM, THRU, NPV, IRR and FOR. Reserved words are auto completed after typing in one or more of the first two characters of a word. An equation can refer to variables in any other node such as Total Expenses, as shown.

The node source is referenced by the reserved word IN. Nodes can be added, renamed or deleted at any time.

The column statement window shows columns for years 2005, 2006, 2007 and Total. The special column window shows computed columns such as summing columns one through three and calling it Total, as in this example. There can be any number of special columns.

Figure 22.3. A sample Planners Lab model

Figure 22.4. Display of model errors after clicking on Validate

Clicking the Validate button checks the entire model for syntax and logic errors. If there are errors, the node name, equations and highlighted specific errors are displayed. The user can also check for syntax errors in a selected node at any time. Figure 22.4 is an example of results from validation. Note that errors are highlighted in yellow in the equations themselves

When clicking on Solution Views, one user option is Tables (standard spreadsheet); here, what-if changes and goal-seek questions can be pursued. Another option is Trend Lines; the user may drag and drop points to pursue what-if changes and ask goal seek questions. Other current options in the program include charts using the Xcelsius software.

Animation is used extensively throughout the Planners Lab software. *It is simple to implement animation with the Flash development platform.* For example, the red borders pulse around nodes with errors to provide a visual cue of an item that needs attention.

Figure 22.5 is an example created by clicking on Tables in Solution Views. The user is prompted to select columns and variables to be displayed in the spreadsheet. The table is dynamically created as the user picks columns and then variables for row titles from various nodes. Upon completion, the user can select to ask what-if and goal-seek questions by clicking on the appropriate button in the lower right corner.

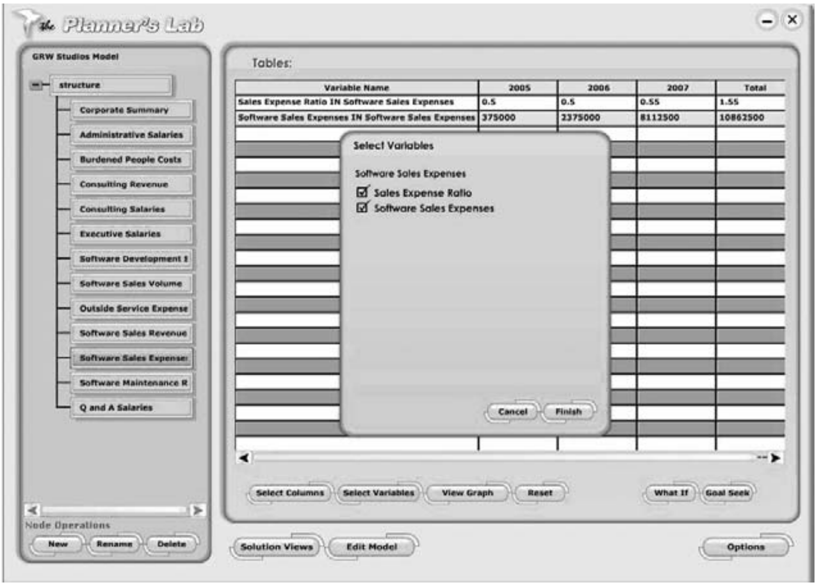


Figure 22.5. Picking column names and variable names for tables

If What If is selected with the Tables option, the user simply clicks on nodes to open it. Then, the user may edit the equations to make temporary changes followed by solving the model (Figure 22.6). Any number of changes can be added cumulatively until the user clicks on Clear.

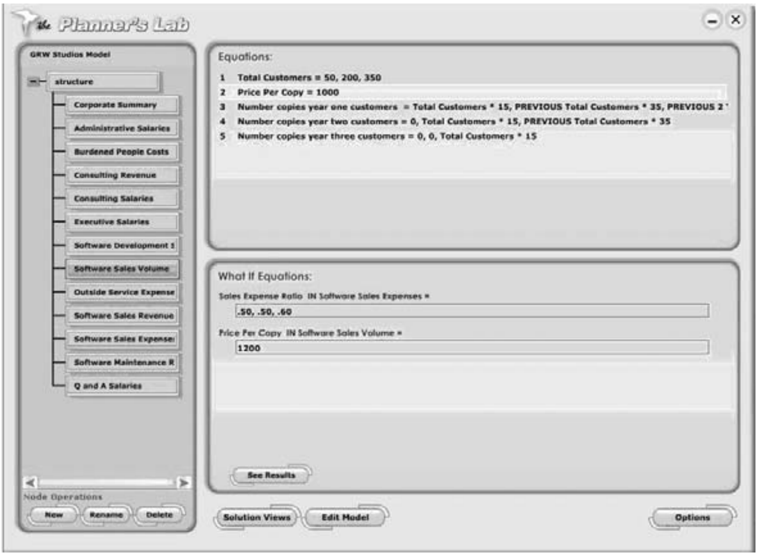


Figure 22.6. Entering what-if questions

Figure 22.7 illustrates the Goal Seek feature with the Tables option. The user clicks on a node from which to choose a goal variable and a node from which to select a changeable variable. In Goal Seek, users indicate what they want the goal variable to achieve and which what-if variable to move to achieve that goal value

Figure 22.7. Entering Goal Seek questions

Another solution view option is Trend Lines (Figure 22.8). In this mode, the user goes through the same process as with tables to select variables.

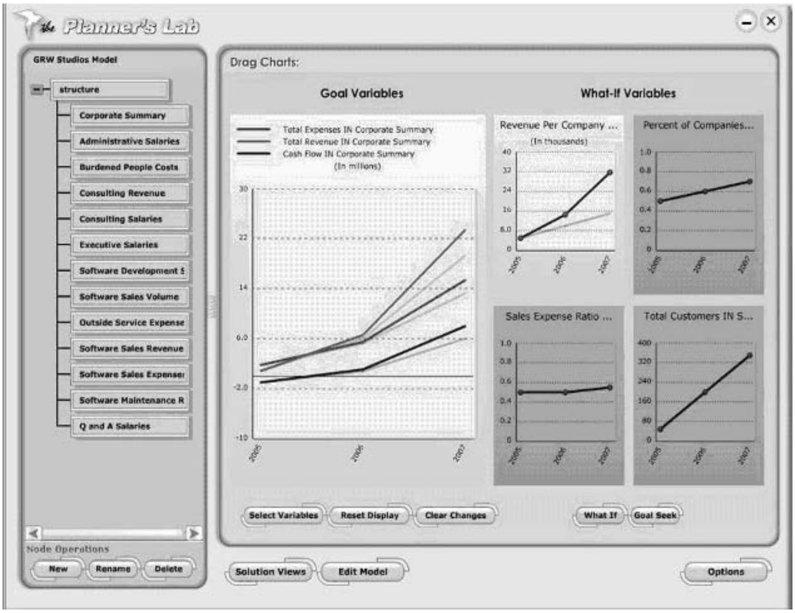


Figure 22.8. What-if analysis within trend lines

The result is a window on the left that shows the goal variables and a window on the right that shows the what-if variables. If What If is selected, the user drags and drops the line for a what-if variable. The goal variables, if affected, are then automatically updated. The lighter color shadow lines show the most recent values for variables prior to what-if or goal seek. What-if changes are not accumulated while moving from one what-if variable to another.

In a similar manner, the user can click Goal Seek and then drag one goal variable line to a desired value point as shown in Figure 22.9. Each what-if variable then adjusts as needed to achieve the goal. Each what-if variable behaves independently of all others.

The Excel spreadsheet generated from the Planners Lab can be used in conjunction with Xcelsius (discussed previously in this chapter) to provide interactive and highly attractive charts. The example in Figure 22.10 combines a dashboard look and feel-along with animated what-if slider bars.

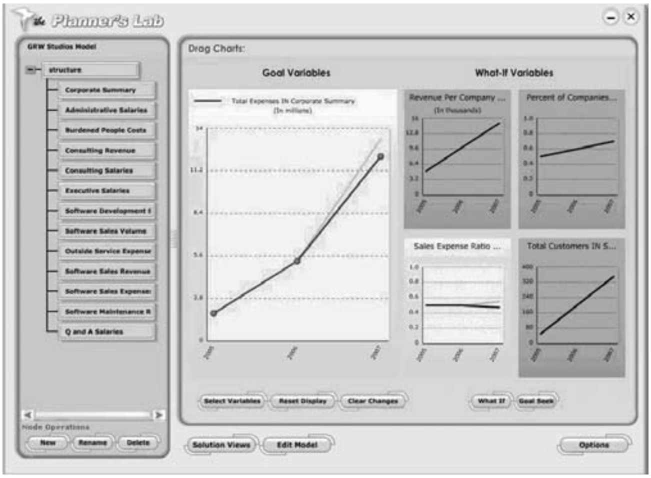
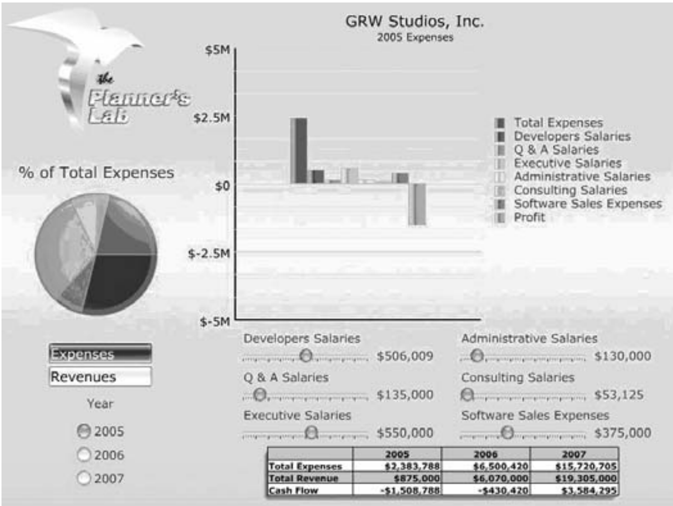


Figure 22.9. Goal Seek analysis within trend lines



	2005	2006	2007
Total Expenses	$2,383,788	$6,500,420	$15,720,705
Total Revenue	$875,000	$6,070,000	$19,305,000
Cash Flow	-$1,508,788	-$430,420	$3,584,295

Figure 22.10. An Xcelsius chart using Planners Lab data

22.4 Academic Collaboration

The software is available to academic institutions for use in teaching, research, and for faculty and student consulting. Campus-wide licenses are available for no up-front cost. They only pay a reduced academics maintenance fee that can also be paid with acceptable "goods in kind," such as an acceptable published paper, chapter or a student thesis. Student consultants are available for onsite training for a modest honorarium plus travel expenses.

The developers of the Planners Lab will have no-cost, nonexclusive rights to use any intellectual property created by academic collaborators.

An annual users-group conference will be held for teaching and research professors and their students to gather and present papers on innovative applications and/or extensions of the Planners Lab software. There will be a peer-review process for accepting papers for presentation; all presented papers will be published in annual proceedings. Releases will be given for those authors whose papers are accepted for publication in professionally refereed journals.

22.4.1 Examples of R&D Projects

The software may be used as a laboratory to support research on decision making and a search for improvements in decision making processes. The following are example research areas consistent with the overall vision of the Planners Lab:

22.4.1.1 Experimentation on the Psychology of Perceptions

Generation and examination of alternative futures using advanced visualization technologies has enormous potential if guided by a consideration of human capacities and preferences. A focus on the user *perceptions* should be included in all experiments. Specifically, the design and assessment of software can include empirical evaluation of the interaction appeal and the acceptance of the simulation results. Acceptance of the modeling results is critical. Projected outcomes must be credible.

A simulation's credibility may take one of four forms: 1) presumed; 2) surface; 3) reputed; or 4) earned. This research can evaluate software on all four of these facets, the most critical of which is the last, which reflects the firsthand experience of the user. Such evaluations will lead to important software enhancements.

22.4.1.2 Viewing Simulator Results as Maps

This research can bring to bear expertise with geospatial data, visualization, interactive maps and animation technology for including *location components* into DSS. Visualization and geographical information technologies for mapping are well established for physical entities but are rarely a part of real-time business simulators for rehearsing the future.

A single data set from one set of changed assumptions can yield maps that are extremely variable, in a visual sense, depending on the mapping methods selected. Because most existing mapping software offers multiple methodological choices, it stands to reason that decisions may depend on the choices made at the time of map production.

The goal of this research is to develop tools that will permit the decision maker to view simulator results from different mapping choices, by viewing a multitude of maps in animation sequences. The maps would be produced by accessing a database and choosing a range of mapping settings, automatically making one map at a time. Finally (and automatically), each map would be inserted as an animation frame for playback and inspection. Thus the decision maker can view results from a simulator with many maps, rather than a single map.

22.4.1.3 Visualizing Temporal Issues in Simulator Results

The ThemeRiver visualization software was founded by the Pacific Northwest National Laboratory (Havre *et al.* 2000) to help identify *time-related* themes, patterns, trends and relationships in collections of documents. Simulated data produced from the Planners Lab software is almost always time-series data.

A collection of documents is represented as a "river" in ThemeRiver; currents flow through time and the river's overall width changes to depict changes in the collective strength of selected themes in underlying documents. Individual themes are represented as colored "currents" flowing within the river. These currents narrow or widen to indicate decreases or increases in the strength of individual themes at any point in time.

22.4.1.4 Intelligence Amplification – Augmenting Minds

The Planners Lab software implements a philosophy of intelligence amplification rather than artificial intelligence (Brooks 1996). Intelligence amplification is the augmentation of minds providing the means for better understanding, judgment, intuition, inference and meaningful action.

Intelligence amplification's research seeks to answer key questions, including: 1) What is the extent to which the human intellect improves and expands through the application of simulators and visualization?; and 2) How do these tools improve problem solving and artistic expression?

22.4.1.5 Incorporating Storytelling into Decision Support

Storytelling was an innate part of ancient cultures. It is the core of Broadway plays, movies and novels. In their prescient and oft-cited 1973 article on a "Program of Research for MIS," Mason and Mitroff suggested that storytelling might be useful as an output medium for information systems. As they put it:

> *"In sharp contrast to the impersonalistic nature of formal models, computer printouts and displays and company reports is the alternative of a more personalistic approach. Stories, drama, role plays, art graphics, one-to-one contact and group discussion may be more effective in some information contexts and, if so, they may suggest media, channels and technologies for presentation which are radically different from the computer and its ancestors – manual accounting systems, adding machines, bookkeeping machines and punch cards. Television, radio, films and telephones may begin to take on a more important role in the MIS of the future."*

Perhaps they could not have envisioned that all the technologies they mention are now a part of present-day computers. Yet, to our knowledge, the use of such multimedia technology to support decision making has been investigated very little.

How can storytelling know-how be applied to DSS for the plan builder to write a better story via the simulator? How can the story be communicated to others more quickly, easily and accurately? The collection of assumptions in a DSS tells a story about a plan, a strategy, a budget or a tactic. The user interacts with the story as in a focused conversation.

22.4.1.6 Semiotics and Decision Support

Semiotics is defined as the science or doctrine that studies signs (Danesi 1994). It is concerned with how minds produce and communicate meaning through symbols. Semiotics considers anything representative—such as a word or gesture—to be a sign. There is no creation of communication or meaning if information is not perceived thorough our senses. The challenge with technologies for DSS and visualization is to have the best "signs" formed from a combination of codes, media and contexts to make meaning fast, cost effective and accurate.

This research can explore how the theory of semiotics can be a basis for enhancing the conversations between DSS and decision makers. These conversations are filled with symbols or signs such as error messages, graphics, charts, tables, colors, shapes, motion, check boxes, drop down menus, radio buttons, task bars and other signs.

22.4.2 Specific Questions to Guide Research

Following are example questions that the software aims to answer with the involvement of its academic collaborators:

- How can we, in real time, change decision assumptions and provide engaging experiences similar to those experienced by electronic game players who receive immediate and visual feedback?

- How can we see more engaging, insightful and animated information rather than static bar charts, line charts and pie charts?

- How can we "fly through" our simulated business plans watching an interactive movie and have experiences similar to that of a civil engineer watching a movie of simulated alternative bridge designs?

- How can we view realistic animations versus spreadsheets for alternative investment strategies?

- How can we read the assumptions in a simulation model as though it were another kind of story? Is storytelling more compelling as an output medium than typical tables, charts and graphs?

22.5 Training Innovative Software Developers

A new breed of software developer/engineer is needed to create today's DSS. This person will be a combination of artist and computer scientist. The new breed is now clearly inventing and developing today's most innovative software for organizations such as Pixar and Universal Studios. Furthermore, this area is where the growth is in the software industry. It's hoped that the teachers and researchers who become affiliated with the Planners Lab will be cross disciplinary and will include the likes of painters, sculptors, poets, graphics designers and communication specialists. People are now advocating that software development is as much an art as it is a science; it's time for DSS to adequately incorporate what both the artists and the scientists have to offer.

The following is an excerpt from an essay by Dr. Paul Graham called "Hackers and Painters" and is posted on his Web site: **www.paulgraham.com/hp.html.**

> *"What hackers and painters have in common is that they're both makers. Along with composers, architects, and writers, what hackers and painters are making are things. They're not doing research per se, though if in the course of trying to make good things they discover some new technique, so much the better."*

Dr. Fred Brooks had the following to say in his acceptance speech as the first recipient of the ACM Allen Newell Award:

> *"I submit that by any reasonable criterion the discipline we call 'computer science' is in fact not a science but a synthetic, an engineering, discipline. We are concerned with making things, be they computers, algorithms, or software systems…In contrast with many engineers who make houses, cars, medicine, and clothing for human needs and enjoyment, we make things that do not themselves directly satisfy human needs, but which others use in making things that enrich human living. In a word, the computer scientist is a toolsmith - no more, but no less. It is an honorable calling."*

Dr. Rob Austin and Dr. Lee Devin in their new book, *Artful Making - What Managers Need to Know About How Artists*, have this to say:

> *"Artful making (which includes agile software development, theatre rehearsal, some business strategy creation, and much of other knowledge work) is a process for creating form out of disorganized materials. Collaborating artists, using the human brain as their principal technology and ideas as their principal material, work with a very low cost of iteration. They try something and then try it again a different way, constantly re-conceiving ambiguous circumstances and variable materials into coherent and valuable outputs."*

The following is an excerpt from an essay by Dr. Richard P. Gabriel who also proposes the idea of a Master of Fine Arts in Software (www.dreamsongs.com):

> *"Software development is like putting on a play, which requires the skills and performances of a number of people working in tandem on stage and behind the scenes. Such skills can be developed in isolation through practice with other amateurs or even by putting on plays in public without any training at all. But how much faster could talent be developed in an educational program that recognized that writing software has enough of an arts-like performance component that the program was tailored to it?"*

It's clear that, in terms of what's considered a superior product or a model in best practices, the artful creation of serious business software is in a period of change. It's also clear that universities must consider if their curricula and resources are adequate to prepare the brightest and most creative to become makers of innovative software.

22.6 Summary

It's anticipated that the Planners Lab software will be used by hundreds of academic institutions around the world. The Planners Lab is a learning and research platform from which to build towards the vision of business decision makers having decision support simulators that provide engaging and even fun experiences similar to that gained by gamers. Together with its academic associates, the software will advance the state of the art in DSS technology and applications. Advancements in the future are limited only by desires, creativity and imagination.

Clearly the description of the Planners Lab software in this chapter is only an introduction, as the software itself is continually changing. Subsequently, the content of this chapter is likely to be somewhat obsolete the day this chapter is published. The purpose of this chapter, however, is to present concepts and vision rather than complete operating details. A users manual and other details are available upon request from **gerald.wagner@grwstudios.com.**

References

Austin R, Devin L (2003) Artful Making - What Managers Need to Know About How Artists Work. Pearson Education Publishing as Prentice Hall Financial Times.

Brooks F (1996) The computer scientist as tool-smith II. Communications of the ACM, 39(3): 61-68.

Danesi M (1994) Messages and Meaning: An Introduction to Semiotics, Canadian Scholars, ISBN 1551300273.

Graham P (2004) Hackers and Painters. O'Reilly Media, Inc.

Havre S, Hetzler E, Whitney P, Nowell L (2000) ThemeRiver: Visualizing Thematic Changes in Large Document Collections, IEEE Transactions Visualization and Computer Graphics, Vol. 8, No. 1, pp. 9-20

Kosy DW, Wise BP (1983) Self-Explanatory Financial Planning Models, Proceedings of the National Conference on Artificial Intelligence, pp. 176-181

Mason RO, Mitroff II (1973) A Program for Research in Management Information Systems, Management Science, 19(5): 475-485

Panko RR (1998) What We Know About Spreadsheet Errors. Journal of End User Computing. Special issue on Scaling up End User Development 10(2): 15-22

Paradice DB, Courtney Jr. JF (1987) Causal and Non-Causal Relationships and Dynamic Model Construction in a Managerial Advisory System, Journal of Management Information Systems, 3(4): 39-53

23

A Strategic Descriptive Review of the Intelligent Decision-making Support Systems Research: the 1980–2004 Period

Manuel Mora[1], Guisseppi Forgionne[2], Jatinder N.D. Gupta[3], Leonardo Garrido[4], Francisco Cervantes[5] and Ovsei Gelman[6]

[1] Department of Information Systems, Autonomous University of Aguascalientes, Aguascalientes, Mexico, 20000, e-mail: mmora@securenym.net
[2] Department of Information Sy ems, University of Maryland Baltimore County, Catonsville, MD 21250 USA, e-mail: forgionn@umbc.edu
[3] College of Administrative Science, The University of Alabama in Huntsville, Huntsville, AL 35899, USA, e-mail: guptaj@uah.edu
[4] Center for Intelligent Systems, Monterrey Tech, Monterrey, N.L., Mexico, 64800, e-mail: leonardo.garrido@itesm.mx
[5] Center of Applied Sciences and Technology Development, National Autonomous University of Mexico (UNAM), Mexico, D.F. 04510, e-mail: francisco_cervantes@cuaed.unam mx
[6] Center of Applied Sciences and Technology Development, National Autonomous University of Mexico (UNAM), Mexico, D.F. 04510, e-mail: ogelman@servidor.unam.mx

About 25 years ago, the Nobel laureate Herbert A. Simon and other top *Management Science/Operations Research* (MS/OR) and *Artificial Intelligence* (AI) researchers, suggested that an integration of the two disciplines would improve the design of decision-making support tools in organizations. The suggested integrated system has been called an intelligent decision-making support system (i-DMSS). In this chapter, we use an existing conceptual framework posed to assess the capabilities and limitations of the i-DMSS concept, and through a conceptual meta-analysis research of the *Decision Support System* (DSS) and AI literature from 1980 to 2004, we develop a strategic assessment of the initial proposal. Such an analysis reveals support gaps that suggest further development of the initial i-DMSS concept is needed. We offer recommendations for making the indicated improvements in i-DMSS design, development, and application.

23.1 Introduction

The decision-making process, from the phases of *intelligence* for setting the decisional agenda and *design* for representing the decisional problem in a suitable model and discovering feasible alternatives, to the phases of *choice* for evaluation and selection of alternatives, *implementation* of the course of action selected, and *learning* phases based on the decisions taken, has been considered one of the most critical and central activities performed in organizations (Simon 1987, 1997, Huber 1990, Holsapple and Whinston 1996).

It has been also recognized - long ago - in the literature (Howard 1968, Drucker 1967, Sage 1981) and nowadays that the hostile, complex, uncertain, and highly competitive business environment (Huber and McDaniel 1986, Huber 1990, Nolan 1991), and the existence of multiple variables, the presence of uncertain data, the significance of risks of negative outcomes, and the strong organizational, time and economic pressures of conflictive decisional goals, values and preferences, hinder modern decision makers from taking efficient and effective decisions. Consequently, bad executive decisions has been identified as a main reason of executive removal in large organizations (Rowe and Davis 1996).

In order to improve the process and outcome's efficiency and effectiveness, several computerized support tools, collectively called *Decision-making Support Systems* (DMSS), have been proffered from the early 1970s (Gorry and Scott-Morton 1971). These systems include *Decision Support Systems* (DSS), *Executive Information Systems* (EIS), *Expert and Knowledge-based systems* (ES/KBS), and other standalone systems (Forgionne *et al.* 2000). From the literature, Table 23.1 exhibits the main reasons of their utilization.

Table 23.1. Main reasons for using core DMSS

DSS usage Reasons	EIS usage Reasons	ES/KBS usage reasons
• Improve the quality of decisions. • Increase productivity of analysts. • Facilitate communication between decision makers and analysts. • Save analysis time. • Support objective-based decisions. • Reduce costs derived from wrong decisions. • Incorporate decision-maker's insights and judgments into analysis.	• Increased competition. • A highly dynamic business environment. • Need for a fast executive response. • Need for timely executive information. • Need for improved communications • Need for rapid status on operational data. • Scan the external decision environment. • Capture, filter, and focus external and internal data.	• Preserve valuable and scarce knowledge. • Share valuable and scarce knowledge. • Enhance problem-solving abilities of users. • Develop user job skills. • Increase productivity. • Improve quality of solution provided. • Guide the user through the problem- solving process. • Provide explanations for recommended actions.

Furthermore, long ago, a group of *Management Science* and *Artificial Intelligence* (AI) experts leaded by Herbert A. Simon suggested that AI-based techniques could greatly enhance DMSS designs by incorporating more complete representations for data, information and knowledge models and more intelligent processing algorithms than traditional systems (Simon 1987, Simon *et al.* 1987, Jacob *et al.* 1998). Similar claims were posed as a research stream for the *Decision* and *Management Sciences* community (Little 1986). Its research proposal resulted in integrated or hybrid systems that have been called *Intelligent Decision-Making Support Systems* (i-DMSS). Internal and external pressures aforementioned on decision makers and Information Technology (IT) advances fostered their emergence (Eom 1998a).

However, despite individual progress in the development and application of i-DMSS (Goul *et al.* 1992, Eom 1998a), it has been recently reported these efforts still have important knowledge gaps (Mora *et al.* 2003). It has also been hypothesized that it is caused by lack of an interdisciplinary research approach despite the initial proposals to integrate the DMSS and Artificial Intelligence (AI) disciplines (Simon 1987, Simon *et al.* 1987). Consequently, we claim that the i-DMSS concept has not fully realized its promise for users, groups and organizations.

This chapter seeks to establish an overall evaluation of the advances in the field and through the knowledge gaps identified, setting a strategic research agenda to help DMSS and AI community to advance towards the full realization of the i-DMSS concept. Using a conceptual research approach (Alavi and Carlson 1992), and a recently reported framework for classification of the support capabilities provided by all types of DMSS (Mora *et al.* 2003), we report a descriptive strategic review of main i-DMSS literature produced in top MS/OR and AI journals for the 1980-2004 period . Our study differs from previous DSS and AI studies (Elam *et al.* 1986, Eom 1998a, 1988b, *Doyle et al.* 1996) in two important ways: (a) the type of analysis conducted (*i. e.* what phases and steps of the decision-making process have received more research effort by the i-DMSS research community and what AI main mechanisms have been used from 1980 to 2004) and (b) the scope of the literature reviewed (*i. e.* a most representative list of the top 25 journals of both MS/OR and AI fields that offer a world perspective). Other studies have focused on the intellectual structure of the field (*i. e.* the underlying topics mainly studied via cocitation analysis) and/or have reviewed a few journals focused in a region (*i. e.* journals published in either the Americas or Europe). This research extends a similar investigation done exclusively for MS/OR journals of authors (Mora *et al.* 2005a).

Through the strategic review and analysis of i-DMSS papers reported in top MS/OR and AI journals from 1980 to 2004, we revealed support gaps that suggest a consolidated and integrated DMSS and AI research agenda for the next generation of intelligent DMSS. Implications for research and practice will be presented, and further research recommendations will be given.

23.2 Capability Assessment Framework for DMSS (CAF-DMSS)

Previous conceptual studies, based on types of DMSS (Sprague 1980, Rockart and Tracy 1982, Turban and Watson 1994), have viewed the systems along several dimensions. These dimensions include: (i) types of decisions and managerial levels *versus* types of information systems (Turban and Aronson 1998), (ii) types of decision phases or decision steps *versus* types of information systems (Forgionne 1991), (iii) types of decision tasks (Elam and Konsynski 1987), (iv) types of density for the intelligence process (Dhar and Stein 1998), and (v) types of basic and optional structural components *versus* types of information systems (Mentzas 1994). These perspectives have generated valuable findings to help match information systems with required decision-making support.

According to Sprague (Sprague 1980), a conceptual framework can help to organize convergent issues in a discipline or topic by identifying the relationships between parts and revealing missing areas for further research. From this perspective, the previous frameworks are incomplete and nonintegrated for the analysis of i-DMSS. We offer a more detailed focus that allows an assessment of the different types of support provided by AI and non-AI technologies on the decision-making phases and steps. The new conceptual tool developed, the *Capability Assessment Framework for DMSS* (CAF-DMSS) (Mora *et al.* 2003), integrates and extends all previous work and thereby provides a detailed and holistic conceptual schema for the appraisal of structural components and corresponding DMSS capabilities. The CAF-DMSS identifies three dimensions as core structural components: (i) the user-interface capability dimension (UICD), (ii) the data, information and knowledge-representation capability dimension (DIKCD), and (iii) the processing-capability dimension (PCD). The first and second dimensions are based on the general and standard structure for a DMSS (Sprague 1980, Rockart and Tracy 1982, Turban and Watson 1994). The third dimension is based on the types of decisions tasks, levels of intelligence embedded in algorithms, and types of intelligent operations for intelligent data-mining systems suggested in the literature (Elam and Konsynski 1987, Dhar and Stein 1998, Gray and Watson 1996). An ordinal scale, derived from an analysis of previous frameworks, is used to measure the three dimensions.

The UICD dimension is divided into three levels that examine the richness of the presentation and action language. The DIKCD dimension is divided into five levels of structural complexity of schemes of representation. Finally, the PCD dimension, which is divided into five levels, describes the degree of embedded intelligence in the algorithms and heuristics for data, information and knowledge processing. Any support level can include capabilities from the previous level. Tables 23.2, 23.3 and 23.4 present a description of the levels for the UICD, DIKCD and PCD dimensions, respectively.

This multidimensional framework is useful to analyze, with a high degree of conceptual detail, the support capabilities in past, current, and future DMSS (Mora *et al.* 2005a). The framework also can be used to improve and enhance the design of DMSS by detecting topics where further research is required (Forgionne *et al.* 2002a, Mora *et al.* 2003, 2005b).

Table 23.2. Levels of user-interface capability

UICD Levels	Description of Capability
I	The DMSS provides an action language of structured commands and/or menus and a presentation language based on texts and nondynamic or animated graphics.
II	The DMSS provides an action language of structured commands and/or menus and a presentation language based on hypertext, or multimedia graphics, sounds, animations and video or dynamic graphics or simulation-based outcomes.
III	The DMSS provides an action language of natural language and/or a presentation language based on virtual reality environments.

Table 23.3. Levels of da , information and knowledge-representation capability

DIKCD Levels	Description of Capability
I	The DMSS uses plain files, simple data structures or/and one-dimensional database schemes to represent data and information items.
II	The DMSS uses complex and highly structured data structures or/and multidimensional database schemes to represent data and information items.
III	The DMSS accesses structured data, information and knowledge organized in quantitative models, such as forecasting models, simulation models, statistical models, Bayesian networks, and neural layers.
IV	The DMSS accesses highly semistructured data, information, and knowledge organized in knowledge chunks. Examples are if-then rules, if-then fuzzy rules, semantic networks, frames, scripts, and cases.
V	The DMSS accesses a network of highly ill-structured data, information and knowledge organized in knowledge bases. Examples are ontology-based repositories and distributed knowledge bases.

Table 23.4. Levels of processing capabilities

PCD Levels	Description of capability
I	The DMSS provides all SQL-like actions: searching, adding, updating, deleting and sorting using a crisp logic mechanism. Also it supports all OLAP-alike actions such as drilling-down rolling-up, slicing and pivoting operations.
II	The DMSS provides all OLAP-alike actions: drilling-down, rolling-up, slicing and pivoting operations and/or all SQL-like actions of searching, adding, updating, deleting and sorting for fuzzy data.
III	The DMSS provides services of classification, association, clustering, trend analysis and forecasting for quantitative data. Examples are neural networks, genetic algorithms, data mining and statistical-based algorithms.
IV	The DMSS provides services of algorithms and heuristics for com lex analysis tasks with both qualitative and quantitative data, such as classification, diagnosis, interpretation and monitoring/control. Examples are rule-based inference algorithms, case-based techniques, and frame and semantic networks inference algorithms.
V	The DMSS provides services of algorithms and heuristics for complex synthesis tasks with both qualitative and quantitative data, such as discovering, explanation, planning, design and learning. Examples are agent-based behavioral algorithms and hybrid or integrated intelligent algorithms.

In this work, we will focus our analysis on the achievements, limitations, and trends of intelligent DMSS research for the development of a strategic research agenda.

23.3 Research Methodology

We adopted the utilization of a conceptual meta-analysis research approach – *i. e.* a systematic research review - used similarly in *Decision Support Systems*, *Information Systems* and *Software Engineering* research (Elam *et al.* 1986, Alavi and Carlson 1992, Morrison and George 1995). According to a recent research approach and research method classification system (Glass *et al.* 2004), our study can be considered a piece of *descriptive research approach* with a *literature review research method*. A set of theoretical and empirical works published from 1980 to 2004 in well-recognized journals in the fields of *Decision Support Systems* (DSS) and *Operations Research/Management Sciences* (OR/MS) as well as in the *Artificial Intelligence* discipline were thoroughly reviewed. The list of journals was selected in three steps.

First, we generated a ranked list of the best 30 journals, using the following selection criteria: (i) the journal appeared in well-known ranked lists (Eom 1998b, Saunders 2004, Forgionne *et al.* 2002b, Cheng *et al.* 1994, Cheng *et al.* 1996); (ii) the journal was recommended by DMSS and AI academic associations, such as SIGDSS, OR and AIS, ACM, IEEE or AAAI, and (iii) the journal was suggested by the authors' joint expertise as being relevant to DSS and AI literature simultaneously. In the second step, we searched the best 25 journals for relevant theoretical and empirical works on intelligent DMSS using several online abstract search services - EBSCO, SciencesDirect, DecisionWeb, ACM and IEEE Digital Library -. In the search, we used the following phrases: (intelligent OR intelligence OR expert OR knowledge OR "artificial intelligence" OR AI) AND ("decision support" OR "decision making" OR "decision maker" OR "decision problem" OR DSS). Two independent codifiers read extended abstracts and selected a total of 485 and 463 papers from DMSS and AI journals, respectively. In the third and final step, the original ranks of the best 30 journals were weighted by the quantity of papers reported in every journal to generate a final weighted-ranked list of 25 DMSS journals and 29 AI journals. The authors agreed on the selection for at least 95% of the population of the initial 485 and 463 papers. At the conclusion of this step, we had a list of 473 DMSS and 465 AI papers from the best 25 and 27 DMSS and AI journals in the weighted-ranked list.

The period of analysis was divided from 1980 to 1989 and from 1990 to 2004 for the following reasons: (i) to facilitate the comparative analysis of data; (ii) to differentiate the different growing stages posed by the generic model of a S-Curve (from initial, contagion, and control to a maturity stage), where in the initial stage of decade of 1980-1989 generated it was cumulated among the 15% to 20% of the total papers (the initial stage) and (iii) because seminal papers from top senior leaders that shaped the field were reported in this initial decade. Tablea 23.5 and 23.6 exhibits the weighted-ranked lists for the DMSS and AI journals, respectively.

In order to integrate the database of DMSS and AI papers, the codification process of each paper was developed using a pro-forma abstract technique (Baidoo *et al.* 2004). The attributes of the pro-forma were selected with the purpose to collect data useful for answering the key underlying research question of this work: what have been the main achievements of AI-based mechanisms, models and paradigms in the improvement of the design of i-DMSS as well as in the process -phases and steps - and outcomes of decision-making?. Table 23.7 exhibits this pro-forma.

Table 23.5. W-ranked list of top DSS and OR/MS journals

ID-Code	Short-Name	Journal	Rank	W-Rank	Number of Papers	Papers 1980-1989	Papers 1990-2004	Percent of i-DMSS research	Accumulated Percent
1	DSS	Decision Support Systems	8.5	2.65	151	31	120	31.1%	31.1%
2	EJOR	European Journal of Operational Research	5.5	0.62	55	10	45	11.3%	42.5%
3	DS	Decision Sciences (*)	8.5	0.37	21	0	21	4.3%	46.8%
4	INT	Interfaces	7.5	0.36	23	7	16	4.7%	51.5%
5	CIE	Computer and Industrial Engineering	3.5	0.32	45	11	34	9.3%	60.8%
6	JMIS	Journal of Management Information Systems	4.5	0.29	31	10	21	6.4%	67.2%
7	MS	Management Science	7.5	0.28	18	10	8	3.7%	70.9%
8	JORS	Journal of Operations Research Society	4.0	0.26	32	0	32	6.6%	77.5%
9	OR	Operations Research	7.0	0.17	12	6	6	2.5%	80.0%
10	JDS	Journal of Decision Systems	5.5	0.14	12	0	12	2.5%	82.5%
11	IPM	Information Processing & Management	4.5	0.10	11	2	9	2.3%	84.7%
12	MISQ	MIS Quarterly	4.5	0.07	8	3	5	1.6%	86.4%
13	AOR	Annals of Operations Research	3.5	0.07	10	0	10	2.1%	88.5%
14	ISR	Information Systems Research	3.5	0.05	7	0	7	1.4%	89.9%
15	ITOR	International Transactions in Operational Research	3.0	0.06	10	0	10	2.1%	92.0%
16	IM	Information & Management	6.0	0.05	4	0	4	0.8%	92.8%
17	OM	Omega	6.5	0.05	4	0	4	0.8%	93.6%
18	JITDM	Journal of Information Technology & Decision Making	2.5	0.05	9	2	7	1.9%	95.5%
19	DBAIS	The Data Base for Advances in IS	5.0	0.03	3	0	3	0.6%	96.1%
20	IRMJ	Information Resources Management Journal	2.5	0.03	5	0	5	1.0%	97.1%
21	GDN	Group Decision and Negotiation	2.5	0.03	5	0	5	1.0%	98.1%
22	HBR	Harvard Business Review	3.5	0.02	3	2	1	0.6%	98.8%
23	EJIS	European Journal of Information Systems	2.5	0.02	3	0	3	0.6%	99.4%
24	IIE-T	IIE Transactions	3.5	0.01	2	0	2	0.4%	99.8%
25	DA	Decision Analysis	2.5	0.01	1	0	1	0.2%	100.0%
		AVERAGE AND TOTALS	4.7	0.24	485	94	391	100.0%	100.0%

Table 23.6. W-ranked list of top AI journals

ID-Code	Short-Name	Journal	Rank	W-Rank	Number of Papers	Papers 1980-1989	Papers 1990-2004	Percent of i-DMSS research	Accumulated Percent
1	ESWA	Expert Systems with Applications	5.5	2.96	252	0	252	53.7%	53.7%
2	ES	Expert Systems	4.5	0.66	69	0	69	14.7%	68.4%
3	ML	Machine Learning	4.0	0.16	19	4	15	4.1%	72.5%
4	AIMED	Artificial Intelligence in Medicine	2.5	0.13	25	0	25	5.3%	77.8%
5	CACM	Communications of the ACM	4.0	0.11	13	6	7	2.8%	80.6%
6	AIM	AI Magazine	4.0	0.09	11	3	8	2.3%	82.9%
7	TSMC	IEEE Transactions on Systems, Man and Cybernetics (A,B,C)	5.5	0.06	5	0	5	1.1%	84.0%
8	AIENG	Artificial Intelligence in Engineering	2.5	0.04	8	1	7	1.7%	85.7%
9	IJES	Int. Journal of Expert Systems	3.5	0.04	5	0	5	1.1%	86.8%
10	IJIS	International Journal of Intelligent Systems	3.5	0.04	5	0	5	1.1%	87.8%
11	AAI	Applied Artificial Intelligence	2.5	0.04	7	0	7	1.5%	89.3%
12	JAIR	Journal of Artificial Intelligence Research	2.5	0.04	7	0	7	1.5%	90.8%
13	AI	Artificial Intelligence	5.5	0.04	3	0	3	0.6%	91.5%
14	AIE	AI Expert	4.0	0.03	4	0	4	0.9%	92.3%
15	AIC	AI Communications	2.5	0.03	6	0	6	1.3%	93.6%
16	TEM	IEEE Transactions on Engineering and Management	2.5	0.03	6	0	6	1.3%	94.9%
17	IEEE-IS	IEEE Intelligent Systems (former IEEE Expert)	5.5	0.02	2	0	2	0.4%	95.3%
18	IEEE-KDE	IEEE Transactions on Knowledge and Data Engineering	2.5	0.02	4	0	4	0.9%	96.2%
19	IEEE-NN	IEEE Transactions on Neural Networks	3.0	0.02	3	0	3	0.6%	96.8%
20	JACM	Journal of the ACM	4.0	0.02	2	0	2	0.4%	97.2%
21	JIFS	Journal of Intelligent and Fuzzy Systems	2.5	0.02	3	0	3	0.6%	97.9%
22	JIIS	Journal of Intelligent Information Systems	2.5	0.02	3	0	3	0.6%	98.5%
23	IEEE-FS	IEEE Transacctions on PAMI	3.0	0.01	2	0	2	0.4%	98.9%
24	IEEE-TR	IEEE Transactions on Robotics and Automation	2.5	0.01	2	0	2	0.4%	99.4%
25	IJUCS	Int. Journal of Human-Computer Studies	3.5	0.01	1	0	1	0.2%	99.6%
26	IEEE-IP	IEEE Transactions on Image Processing	2.5	0.01	1	0	1	0.2%	99.8%
27	TOE	IIE Transactions on Operations Engineering	2.5	0.01	1	0	1	0.2%	100.0%
28	IEEE-C	IEEE Computer	3.5	-	0	0	0	0.0%	100.0%
29	IJAES	Int. Journal of Applied Expert Systems (CLOSED)	3.5	-	0	0	0	0.0%	100.0%
		AVERAGE AND TOTALS	3.3	0.03	469	14	455	100%	100.0%

Table 23.7. Pro-forma for DMSS papers codification

Item	Values
I.1 Year of publication	• Between 1980 and 1989 • Between 1990 and 2004
I.2 Type of research outcome (based on March & Smith's taxonomy (1995))	• System or prototype (INSTANTATION) • Framework, Paradigm, Model or Architecture (MODEL) • Methodology, Guidelines or Design Approach (METHOD) • Survey, Conceptual Analysis or Comparison of Algorithms (SURVEY) • Design of a new algorithm or heuristic (CONSTRUCT)
I.3 Decision-making phase and step	• Intelligence (Data gathering; Problem recognition) • Design (Model Formulation/Selection; Model Analysis) • Choice (Generation/Evaluation; Selection) • Implementation (Presentation of Results, Task Planning; Task Monitoring) • Learning (Outcomes/Process knowledge analysis; Outcomes/Process knowledge synthesis)
I.4 Capability based on the CAF-DMSS Framework.	• UCD (Level I, Level II, Level III) • DIKCD (Level I, Level II, …, Level V) • PCD (Level I, Level II, …, Level V)
I.5 Main Artificial Intelligence mechanism used.	• Classic Rule-based Systems • Fuzzy Logic • Neural Networks (any architecture) • Cased-Based Reasoning • Genetic Algorithms • Data Mining / Induction Trees • Bayesian/Belief Networks, Decision Trees or Influence Diagrams • Natural Language Processing Mechanisms • Intelligent Agents (single or multisystem) • Other AI technique or not mentioned (Blackboard, Rough Sets, Decision Lists, Hybrid MCD & AI Approaches)
I.6 Area	• Management, Financial, Economy, Marketing or General Business (MFE) • Manufacturing, Engineering or Science (MES) • Government, Transportation, Environment, Real State, Law or Education (GTR) • Health Care (HEA) • Military (MIL) • Computers Science, IS/IT or Software Engineering (CIT) • Decision Support Systems Field (DSS) • Other (OTH)

Table 23.7. continued

I.7 Specific Task	• Task in MFE areas • Task in MES areas • Task in GTR areas • Task in HEA area • Task in MIL area • Task in CIT areas • Task in DSS area • Task in OTH areas.
I.8 Level of reliability of codifier	• Very high reliability (5) • High reliability (4) • Sufficient reliability (3) • Low reliability (2) • Very low reliability (1)

23.4 Intelligent DMSS Assessment

The 473-row and 465-row x multi-8-category tables of collected information, which is not reported here due to space limitations, offered a rich data source to generate the strategic research agenda for the field of i-DMSS. To focus the data on pertinent issues, we calculated descriptive statistics for the collected information. One issue of interest is the distribution of research by AI technique and area cluster. These results are shown in Tables 23.8 and 23.9 for DMSS and AI journals, respectively.

As table 23.8 indicates, during the 1980-1989 period, the majority of papers in DMSS journals reported the utilization of uncommon AI techniques (other AI techniques are in Table 23.7). A similar situation occurred in the 1990-2004 period. This rare situation in DMSS journals can be explained by the fact of the AI specific techniques were not reported in the 92 and 381 abstracts consulted. In contrast with the 465 abstracts from AI journals, it is clear that for the DMSS community the realm is focused in the decision-making process rather than in the development of a specific AI mechanism. However, of the AI techniques reported it is clear that *Rule-based Systems* (RBS) and *Bayesian/Belief Nets* were the most used in the initial period and practically none other AI mechanism was used. In contrast in the next period, *Neural Nets* (NN), *Fuzzy Logic* (FL) and *Intelligent Agents* (IAG) have emerged despite of the leadership still of RBS. *Data Mining* (DM), *Genetic Algorithms* (GA) and *Case-based Reasoning* (CBR) are scarcely used but these are being considered recently.

As an interesting research gap, Table 23.8 clearly exhibits that *Natural-Language Processing Mechanisms* (NPL) have received little academic interest in the i-DMSS field in the full 25-year period analyzed. Furthermore, the 5% of *Fuzzy Logic Systems* reported in the period of 1980-1989 – and not a zero per cent- can be caused because FL have been available from the mid-1960s. In particular, it is worthy to note that a seminal paper from the creator of FL theory, reported in a top DMSS journal (Bellman and Zadeh 1970) was ignored by the DMSS literature. Table 23.8 also shows that the main research has been for the sake of the DSS area in both periods. Regarding specific clusters of domain applications MFE and MES, in that order, were the most popular. Little research has been reported in other clusters except by 9% of papers in the GTR domain.

In contrast with DMSS literature, Table 23.9 shows two interesting findings: (i) the AI Literature ignored the DMSS field in the initial period, by the scarce number of papers found (13 papers *versus* 92 in DMSS literature in the same period) and (ii)

Table 23.8. Distribution of percentage of papers by AI technique and cluster of area founded in DMSS/OR/MS journals

	1980-1989									1990-2004								
	CLUSTER OF AREAS									CLUSTER OF AREAS								
AI TECHNIQUE	MFE	MES	GTR	HEA	MIL	CIT	DSS	OTHER	TOTAL	MFE	MES	GTR	HEA	MIL	CIT	DSS	OTHER	TOTAL
PAPERS	15	16	1	2	1	2	55	0	92	91	55	35	16	13	9	160	2	381
%	16%	17%	1%	2%	1%	2%	60%	0%	100%	24%	14%	9%	4%	3%	2%	42%	1%	100%
Rule-based System	7%	8%	1%	0%	0%	1%	12%	0%	28%	6%	4%	2%	1%	1%	1%	6%	0%	20%
Fuzzy Logic	0%	0%	0%	0%	0%	0%	5%	0%	5%	2%	2%	2%	1%	0%	0%	2%	0%	9%
Neural Networks	0%	1%	0%	0%	0%	0%	0%	0%	1%	7%	1%	1%	1%	0%	0%	3%	0%	12%
Case-Based Reasoning	0%	0%	0%	0%	0%	0%	0%	0%	0%	1%	1%	0%	0%	0%	0%	1%	0%	3%
Genetic Algorithms	0%	1%	0%	0%	0%	0%	0%	0%	1%	1%	1%	1%	0%	0%	0%	1%	0%	4%
Data Mining / Ind. Trees	0%	0%	0%	0%	0%	0%	0%	0%	0%	1%	1%	0%	0%	0%	1%	3%	0%	5%
Bayesian /Belief Nets	1%	1%	0%	2%	0%	1%	7%	0%	12%	1%	0%	1%	1%	1%	0%	4%	0%	8%
NPL Techniques	1%	0%	0%	0%	0%	0%	0%	0%	1%	0%	0%	0%	0%	0%	0%	1%	0%	1%
Intelligent Agents	0%	0%	0%	0%	0%	0%	0%	0%	0%	2%	1%	1%	0%	1%	0%	3%	0%	8%
Other AI Techniques	9%	7%	0%	0%	1%	0%	39%	0%	55%	6%	6%	2%	2%	1%	1%	22%	0%	40%

Table 23.9. Distribution of percentage of papers by AI technique and cluster of area founded in AI journals.

	1980-1989									1990-2004								
	CLUSTER OF AREAS									CLUSTER OF AREAS								
AI TECHNIQUE	MFE	MES	GTR	HEA	MIL	CIT	DSS	OTHER	TOTAL	MFE	MES	GTR	HEA	MIL	CIT	DSS	OTHER	TOTAL
PAPERS	0	1	1	1	0	1	9	0	13	55	80	45	51	10	16	176	19	452
%	0%	8%	8%	8%	0%	8%	69%	0%	100%	12%	18%	10%	11%	2%	4%	39%	4%	100%
Rule-based System	0%	8%	8%	0%	0%	8%	38%	0%	62%	4%	10%	5%	4%	1%	1%	12%	2%	38%
Fuzzy Logic	0%	0%	0%	0%	0%	0%	0%	0%	0%	1%	2%	2%	2%	0%	0%	2%	0%	10%
Neural Networks	0%	0%	0%	8%	0%	0%	8%	0%	15%	2%	2%	2%	3%	0%	0%	5%	0%	13%
Case-Based Reasoning	0%	0%	0%	0%	0%	0%	8%	0%	8%	2%	1%	1%	1%	0%	0%	3%	0%	8%
Genetic Algorithms	0%	0%	0%	0%	0%	0%	0%	0%	0%	1%	2%	1%	0%	0%	0%	2%	0%	5%
Data Mining / Ind. Trees	0%	0%	0%	0%	0%	0%	0%	0%	0%	2%	1%	0%	1%	0%	0%	4%	0%	9%
Bayesian /Belief Nets	0%	0%	0%	0%	0%	0%	23%	0%	23%	0%	0%	1%	1%	0%	0%	6%	1%	10%
NPL Techniques	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	2%	0%	2%
Intelligent Agents	0%	0%	0%	0%	0%	0%	0%	0%	0%	2%	1%	1%	0%	0%	1%	3%	0%	8%
Other AI Techniques	0%	8%	0%	0%	0%	0%	0%	0%	8%	1%	3%	2%	2%	1%	1%	7%	1%	16%

the AI community is clearly aware of the AI mechanisms used (the percentage of other AI techniques or not reported is very low (8%) during the 1980-1989 period and the integration/adaptation of several AI techniques suggests an increasing percentage reported (16%) in next period). Despite the low number of papers in the first period, a similarity with DMSS journals occurs where RBS and *Bayesian/Belief Nets* appear as the main AI mechanisms reported. However, research using FL, GA, DM and IAG to enhance DMSS was not existent for the AI community. In contrast, NN and CBR mechanisms appeared for first time related with i-DMSS. Table 23.9 also exhibits evidences that from 1990 to the present (2004) the AI community has been very active in i-DMSS research. Besides RBS as the main AI mechanism, practically all other main AI techniques have been researched. It is interesting to note that FL and IAG emerge in similar percentages of utilization as other AI techniques. As a similar knowledge gap, it is clear that as the AI as well the DMSS community have ignored research efforts oriented to improve i-DMSS through the NLP mechanisms. In short, despite i-DMSS were little studied by AI community during the initial period of 1980-1990 in the next period both communities are similar in research efforts in the field.

From raw data tables, other relevant evidences collected are exhibited in Tables 23.10 and 23.11. Both tables account for the distribution of frequencies and percentages of papers in DMSS and AI journals classified by the type of research outcome. According to March and Smith's taxonomy (1995), research outcomes can be classified into four types. This research expands March and Smith's taxonomy by including five categories (see Table 23.7 for a brief description of each category). As both tables show, in DMSS and AI journals, research has been concentrated in developing a real or a prototype i-DMSS (*e. g.* "Instantiation"). Other percentages of categories of research outcomes are similar in the full 25-year period for both research streams. A finding for the research agenda is that little research is oriented toward the development of methodologies, guidelines or design approaches.

Table 23.10. Distribution of frequencies and percentages of papers on i-DMSS by type of research outcome found in DMSS/OR/MS journals

		TYPE OF RESEARCH OUTCOME					
		Instantation	Model	Method	Survey	Construct	TOTAL
PERIODS	1980-1989	35	20	7	22	8	92
	1990-2004	182	73	35	52	39	381
	TOTAL	217	93	42	74	47	473
	1980-1989	38%	22%	8%	24%	9%	100%
	1990-2004	48%	19%	9%	14%	10%	100%
	TOTAL	46%	20%	9%	16%	10%	100%

Table 23.11. Distribution of frequencies and percentages of papers on i-DMSS by type of research outcome founded in AI journals.

		TYPE OF RESEARCH OUTCOME					
		Instantation	Model	Method	Survey	Construct	TOTAL
PERIODS	1980-1989	1	1	0	9	2	13
	1990-2004	226	72	45	64	45	452
	TOTAL	227	73	45	73	47	465
	1980-1989	8%	8%	0%	69%	15%	100%
	1990-2004	50%	16%	10%	14%	10%	100%
	TOTAL	49%	16%	10%	16%	10%	100%

The key issue found is the emergence of *Generic-Tasks* (Cuena and Molina 2000, Wong and Bhattacharyya 2002) and *Object-Oriented* (Manjarres and Pickin 2002) approaches to facilitate the design of i-DMSS.

Tables 23.12 and 23.13 exhibit critical evidences for the strategic research agenda. Under the premise that i-DMSS are designed to support and improve the decision-making process and outcomes, DMSS and AI research reported in journals during the 1980-1989 and 1990-2004 periods would show evidences of it.

Table 23.12. Distribution of percentages of papers on i-DMSS by type of decision-making phase and step supported *versus* period found in DMSS/OR/MS Journals.

		DECISION-MAKING PHASES AND STEPS											NUMBER OF PAPERS
PHASES		INTELLIGENCE		DESIGN		CHOICE		IMPLEMENTATION			LEARNING (HUMAN & SYSTEM)		
STEPS		DATA GATHERING	PROBLEM RECOGNITION	MODEL FORMULATION	MODEL ANALYSIS	EVALUATION	SELECTION	RESULTS PRESENTATION	PLANNING	TRACK PLANS	OUTCOME-PROCESS ANALYSIS	OUTCOME-PROCESS SYNTHESIS	
PERIODS	1980-1989	7%	39%	28%	5%	57%	60%	0%	0%	0%	0%	0%	92
	1990-2004	13%	54%	23%	5%	52%	53%	1%	0%	0%	1%	2%	381
	1980-1989	42%		29%		63%		0.0%			0%		92
	1990-2004	60%		23%		60%		0.3%			2%		381

Table 23.13. Distribution of percentages of papers on i-DMSS by type of decision-making phase and step supported *versus* period found in AI journals.

		DECISION-MAKING PHASES AND STEPS											
PHASES		INTELLIGENCE		DESIGN		CHOICE		IMPLEMENTATION			LEARNING (Human & System)		NUMBER OF PAPERS
STEPS		DATA GATHERING	PROBLEM RECOGNITION	MODEL FORMULATION	MODEL ANALYSIS	EVALUATION	SELECTION	RESULTS PRESENTATION	PLANNING	TRACK PLANS	OUTCOME-PROCESS ANALYSIS	OUTCOME-PROCESS SYNTHESIS	
PERIODS	1980-1989	0%	54%	54%	0%	62%	54%	0%	0%	0%	0%	0%	13
	1990-2004	6%	65%	30%	3%	69%	66%	9%	0%	0%	0%	0%	452
	1980-1989	54%		54%		62%		0%			0%		13
	1990-2004	65%		31%		71%		9%			0%		452

The DMSS community has concentrated the research efforts in the improvement of the *Choice* and *Intelligence* phases. Both decision-making phases have been recognized with a lower level of difficulty to be supported than *Design* phase. A comparison of the first and second periods suggests that research to improve the *Design* phase has been exclusively in the step of *Model Formulation*. A similar fact was found in the AI literature, except by a high percentage in the *Model Formulation* step. However, this evidence must be taken with the precaution due that codifiers agreed to classify papers in this category when the paper helped to manually formulate a "*decision model*" (mainly in RBS). Consequently, the *Model Design* phase continues being an important gap research for the DMSS and AI literature.

Another relevant finding to be noted in Tables 23.12 and 23.13 is research reported for the *Implementation* and *Learning* phases in DMSS and AI literature, respectively is void. In contrast, the DMSS and AI communities start to address opposite phases, respectively. In the first case, studies in *Knowledge Management Systems* (KMS) suggest the integration of KMS with DMSS field (Bolloju *et al.* 2002) and in the latter the studies realized in the development of automated explanation facilities support it (Papamichail and French 2003). Hence, the main underlying research question can be partially answered: DMSS and AI research in i-DMSS has been focused on and has improved (based in implicit benefits reported) the *Choice* and *Intelligence* phases of the decision-making process and the outcomes related with their enhancements.

Some specific improvements have been achieved in the *Design* phase and practically null research has been done for the *Implementation* and *Learning* phases. To answer fully the underlying research question, Tables 23.14 and 23.15 for the DMSS journals and Tables 23.16 and 23.17 for the AI journals were generated. The analysis of these tables will be by type of capability according to the CAF-DMSS framework.

From the data of the four tables it is clear that DMSS research has ignored Level II and III of the user-interface capability dimension. A very low percentage (less than 5%) is reported exclusively for Level II - *e. g.* for multimedia and interactive graph simulation-based user interfaces -. The AI community has been more active in the research effort to increase the support level of user interfaces in i-DMSS. However, both research streams have failed to realize the promise of Level III. A relevant research gap in the utilization of NLP and *Virtual Reality* interfaces for i-DMSS exists in the present. In the same dimension - *e. g.* the UICD - in congruence with Tables 23.12 and 23.13 the *Choice* and *Intelligence* phases have received more support that the others. It must be noted, that "Design" has significant support with the codification consideration (*e. g.* not manual support) indicated previously

Regarding the data, information and knowledge-representation capability dimension (DIKCD), percentages of the four tables suggest that both DMSS and AI journals, research reported has been focused on Level III and IV - *e. g.*. structured schemes organized in quantitative models and highly semistructured chunks of data, information and knowledge -. For the first case, the quantitative models are codified using statistical and classic forecasting models, Bayesian graphs and layers of neural networks. In the second one, the main codifications models have been classic if-then rules, fuzzy if-then rules and cases. It is interesting to not in the few studies reported using Level V – *e. g.* distributed schemes and/or knowledge ontologies - is from DMSS – only in the 1990-2004 period - and not from AI journals.

Table 23.14. Distribution of percentages of papers on i-DMSS by type of decision-making phase *versus* capability. (1980-1989) found in DMSS/OR/MS journals.

		Capabilities Assessment Framework for DMSS												
		Types of User-Interface Capabilities			Types of KR Capabilities					Types of Processing Capabilities				
1980-1899	92	Text / Passive Graphics	Multi/Hyper-media / Simulation /Active Graphics	Natural-Alike User Interfaces	Data Base Models	Multi-Dimensional DataBase	Quantitative-Based Models	Knowledge-Based Models	Distributed-KB Models	Conventional SQL + OLAP Tools	Non-Conventional SQL	Quantitative Structured Mechanisms	Symbolic Structured Mechanisms	Problem-Solving for Ill-Structured Situations
INTELLIGENCE	DATA GATHERING	7%	0%	0%	0%	0%	4%	5%	0%	0%	0%	3%	5%	0%
	PROBLEM RECOGNITION	37%	0%	1%	1%	0%	13%	35%	0%	0%	0%	12%	35%	0%
DESIGN	MODEL FORMULATION	25%	2%	0%	0%	0%	16%	20%	0%	0%	0%	16%	20%	0%
	MODEL ANALYSIS	5%	0%	0%	0%	0%	4%	5%	0%	0%	0%	4%	5%	0%
CHOICE	EVALUATION	52%	2%	1%	1%	0%	24%	50%	1%	0%	0%	20%	51%	0%
	SELECTION	54%	2%	2%	2%	0%	27%	52%	1%	0%	0%	23%	53%	0%
IMPLEMENTATION	RESULTS PRESENTATION	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	PLANNING	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	TRACK PLANS	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
LEARNING	OUTCOME-PROCESS ANALYSIS	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	OUTCOME-PROCESS SYNTHESIS	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

Table 23.15. Distribution of percentages of papers on i-DMSS by type of decision-making phase *versus* capability. (1990-2004) found in DMSS/OR/MS journals.

		Capabilities Assessment Framework for DMSS												
		Types of User-Interface Capabilities			Types of KR Capabilities					Types of Processing Capabilities				
1990-2004	381	Text / Passive Graphics	Multi/Hyper-media / Simulation /Active Graphics	Natural-Alike User Interfaces	Data Base Models	Multi-Dimensional DataBase	Quantitative-Based Models	Knowledge-Based Models	Distributed-KB Models	Conventional SQL + OLAP Tools	Non-Conventional SQL	Quantitative Structured Mechanisms	Symbolic Structured Mechanisms	Problem-Solving for Ill-Structured Situations
INTELLIGENCE	DATA GATHERING	11%	1%	1%	1%	1%	6%	7%	3%	1%	1%	6%	7%	2%
	PROBLEM RECOGNITION	48%	4%	1%	1%	0%	30%	35%	6%	0%	0%	29%	36%	6%
DESIGN	MODEL FORMULATION	21%	1%	0%	0%	0%	12%	15%	1%	0%	0%	11%	15%	2%
	MODEL ANALYSIS	4%	1%	0%	0%	0%	3%	4%	1%	0%	0%	3%	4%	0%
CHOICE	EVALUATION	47%	4%	1%	1%	0%	28%	39%	3%	0%	0%	25%	40%	3%
	SELECTION	48%	4%	1%	1%	0%	28%	42%	5%	0%	0%	26%	42%	5%
IMPLEMENTATION	RESULTS PRESENTATION	1%	0%	0%	0%	0%	0%	1%	0%	0%	0%	0%	1%	0%
	PLANNING	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	TRACK PLANS	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
LEARNING	OUTCOME-PROCESS ANALYSIS	1%	0%	0%	0%	0%	0%	1%	0%	0%	0%	0%	1%	0%
	OUTCOME-PROCESS SYNTHESIS	2%	0%	0%	0%	0%	1%	2%	0%	0%	0%	1%	1%	0%

Table 23.16. Distribution of percentages of papers on i-DMSS by type of decision-making phase *versus* capability. (1980-1989) found in AI journals.

		Capabilities Assessment Framework for DMSS												
		Types of User-Interface Capabilities			Types of KR Capabilities					Types of Processing Capabilities				
1980-1899	13	Text / Passive Graphics	Multi/Hyper-media / Simulation /Active Graphics	Natural-Alike User Interfaces	Data Base Models	Multi-Dimensional DataBase	Quantitative-Based Models	Knowledge-Based Models	Distributed-KD Models	Conventional SQL + OLAP Tools	Non-Conventional SQL	Quantitative Structured Mechanisms	Symbolic Structured Mechanisms	Problem-Solving for Ill-Structured Situations
INTELLIGENCE	DATA GATHERING	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	PROBLEM RECOGNITION	54%	0%	0%	0%	0%	31%	31%	0%	0%	0%	23%	31%	0%
DESIGN	MODEL FORMULATION	54%	0%	0%	0%	0%	31%	31%	0%	0%	0%	23%	31%	0%
	MODEL ANALYSIS	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
CHOICE	EVALUATION	62%	0%	0%	0%	0%	31%	54%	0%	0%	0%	8%	54%	0%
	SELECTION	54%	0%	0%	0%	0%	31%	46%	0%	0%	0%	8%	46%	0%
IMPLEMENTATION	RESULTS PRESENTATION	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	PLANNING	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	TRACK PLANS	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
LEARNING	OUTCOME-PROCESS ANALYSIS	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	OUTCOME PROCESS SYNTHESIS	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

Table 23.17. Distribution of percentages of papers on i-DMSS by type of decision-making phase *versus* capability (1990-2004) found in AI journals.

		Capabilities Assessment Framework for DMSS												
		Types of User-Interface Capabilities			Types of KR Capabilities					Types of Processing Capabilities				
1990-2004	452	Text / Passive Graphics	Multi/Hyper-media / Simulation /Active Graphics	Natural-Alike User Interfaces	Data Base Models	Multi-Dimensional DataBase	Quantitative-Based Models	Knowledge-Based Models	Distributed-KB Models	Conventional SQL + OLAP Tools	Non-Conventional SQL	Quantitative Structured Mechanisms	Symbolic Structured Mechanisms	Problem-Solving for Ill-Structured Situations
INTELLIGENCE	DATA GATHERING	4%	2%	0%	0%	2%	5%	5%	1%	2%	2%	4%	4%	1%
	PROBLEM RECOGNITION	66%	8%	2%	2%	3%	46%	50%	3%	3%	3%	41%	50%	3%
DESIGN	MODEL FORMULATION	27%	2%	1%	1%	1%	19%	20%	2%	1%	1%	17%	21%	2%
	MODEL ANALYSIS	2%	1%	0%	0%	0%	1%	2%	0%	0%	0%	1%	2%	0%
CHOICE	EVALUATION	58%	8%	2%	2%	2%	46%	58%	3%	3%	3%	40%	57%	4%
	SELECTION	66%	8%	2%	2%	2%	46%	57%	3%	2%	2%	40%	56%	4%
IMPLEMENTATION	RESULTS PRESENTATION	2%	5%	2%	2%	0%	6%	9%	1%	1%	1%	6%	9%	1%
	PLANNING	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	TRACK PLANS	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
LEARNING	OUTCOME-PROCESS ANALYSIS	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	OUTCOME-PROCESS SYNTHESIS	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

A plausible explanation for it is that the DMSS community is aware of the potential of distributed schemas for knowledge-codification through agent-based systems, while the AI community, if it is true that agent research is vast, its relation with i-DMSS has not been detected as it was with other AI mechanisms - *e. g.* RBS, NN and FL -. In similar situation of the UICD, most support has been focused in *Choice* and *Intelligence* phases. The rise in the 1990-2004 period for AI journals regarding the *Design* phase must be carefully interpreted due to the majority of research effort was being conducted on models, methodologies, surveys or approaches to improve the formulation of a part of the decision model - mainly from Expert Systems research stream - but not for the design of automated tools for this task.

The analysis of the processing capability dimension (PCD) of four tables is congruent with the DIKCD analysis. Level III and IV have been extensively investigated in both the DMSS and AI literature. The main AI processing mechanisms for Level III are quantitative such as: *Neural Networks*, *Classic Statistical-Based Algorithms (Bayesian/Belief Nets*), *Data Mining* and *Genetic Algorithms*, used for classification, clustering, forecasting and trend analysis of data. In the Level IV the main ones were *Rule-Based*, *Fuzzy Logic* and *Case-based inference algorithms*. It must be noted that Level IV mechanisms can process qualitative and quantitative data. In congruence also with DIKCD analysis, the Level V processing mechanisms based in the integration of all AI techniques using agent-based systems has started to be researched in the DMSS literature and to a less extent in the AI field, from an i-DMSS perspective. The *Choice* and *Intelligence* phases are equally the most enhanced decision-making phases.

In the previous four tables, the gray-shaded zones were used in cells to assess the status of the capability support provided by the i-DMSS to specific decision-making steps using the following criteria: (i) to assign a "mature status" – gray level of high intensity - when: (i.1) the technology's capabilities were reported in at least 25% of the papers in the period; (i.2) these technologies have been successful in large real settings and (i.3) the technology is widely available in commercial products; (ii) to assign a "state-of-the-art status" – gray level of low intensity - when: (ii.1) the technology's capabilities were reported in at least 10% but minor to 25% in the papers in the period; (ii.2) these technologies have been successful in some test, real setting, or complex real problem and (ii.32) the technology is available in a few commercial products; and to assign a "research and development status" - gray level of middle intensity - when: (iii.1) the technology's capabilities were reported in at least 5% and at most 10% of the papers in the period; (iii.2) these technologies have been successful in laboratory tests or have shown conceptual evidence of potential capability and (iii.3) the technology is not yet available in commercial products. The rest of the cells used a blank tone.

In summary, from previous findings and with the integration of all evidences collected in this strategic analysis (of 473 DMSS and 465 AI papers using extended abstracts), we can claim it offers useful insights for developing a research agenda for the next generations of i-DMSS. Hence, the underlying research question, partially answered in previous sections, now can be confirmed: DMSS and AI research in i-DMSS has been focused on and has improved (based in implicit benefits reported) the *Choice* and *Intelligence* phases of the decision-making process and the outcomes related to their enhancements. Some specific improvements have been achieved in the *Design* phase and practically no research has been done for the *Implementation* and *Learning* phases. Most AI mechanisms used for this enhancing had been and are: *Rule-Based Systems*, *Neural Nets*, *Fuzzy Logic*, and *Bayesian Nets*. In recent years, the research on *intelligent Agents* has started to be explored mainly by the DMSS community. Relevant research gaps also can be identified from the integration of the analysis of all tables.

23.5 Implications and Conclusions

A number of important findings have been revealed by this research. One deals with the journals publishing i-DMSS research. Three lead journals (Decision Support Systems, Decision Sciences, and the Journal of Management Information Systems) in the first decade (1980-1989) have kept this position in the second period (1990-2004) in the dissemination of high-quality papers on i-DMSS, while publishing other research themes throughout the full spectrum of the DMSS research field. In addition, some older journals, such as the Journal of the Operations Research Society, Computers and Industrial Engineering, and Information Processing & Management, combined with new outlets, such as the Journal of Decision Systems, have contributed strongly to the dissemination effort.

Another finding involves the most popular AI techniques studied and/or applied. Of the wide spectrum of available AI techniques, few are utilized in i-DMSS research. Machine learning (through *Neural Networks*, Genetic Algorithms and

Data-Mining mechanisms) seems to be growing in importance, but still involves few applications. Moreover, some popular AI techniques of the early 1990s, such as *Case-based Reasoning*, seem to have had little impact in the i-DMSS literature. Research is needed to determine whether existing techniques are underutilized or nonintegrated for i-DMSS design, development, and implementation.

There is a related finding for the cluster areas. AI techniques seem to be used mainly for DSS research and in functional (MFE and MES) areas. Service (such as GRT and HEA) and technical (CS/IT) applications have been limited and largely stagnant over time. As with the other areas, these neglected areas could benefit from the support offered by i-DMSS. Perhaps, then, i-DMSS applications to these cluster areas would be a useful research direction for the field.

The type of research outcomes revealed that foundation theories (*i. e.* models) and practical artifacts (*i. e.* instantiations) have been the most popular outcomes. However, research on how to design, build and implement i-DMSS from a more structured and software engineering/systems engineering perspective are still missing in this entire research period (*i. e.* only 9% of the total papers published reported a method as a research outcome). Similar results have been critiqued in the AI community about the lack of well-structured and scientific-based development methodologies (Parnas 1985).

A major deficiency noted in our analysis was the limited support for decision-making phases and steps currently offered by reported i-DMSS. In particular, the design, implementation, and learning phases of the decision-making process receive little research effort and thereby are undersupported in the i-DMSS community. A few exceptions have been focused on model formulation and model analysis in the design phase (Bonczek *et al.* 1981, Elam and Konsynski 1987, Liang 1988, Angerhm 1997, Satty 1998). However, these research efforts continue in a state-of the-art status. Analysis also suggests that overall support within the phases is incomplete and still in the research stage. Since complete and integrated support is crucial to effective decision making, further research is needed to identify methodologies and approaches to expand the breadth and depth of DMP support from i-DMSS.

In the UICD dimension, AI-based advanced techniques of natural language processing and virtual-reality-based interfaces have few reported applications. In the DIKCD and CPD dimensions, the emergence of distributed AI-based representations and mechanism using "intelligent agents" seems to offer an architectural solution for the integration and deployment of the wide spectrum of AI techniques. However, the support offered tends to be qualitative rather than quantitative in i-DMSS. Hence, we suggest that architectures for a best integration of both approaches should be deployed to generate the synergistic benefits. In the period 1990-2004 for example, it seems that analytical capabilities are generally underutilized and AI-based researchers are far from OR/MS foundations, which is contrary to the recommendations of AI and OR/MS leaders, such as H.A. Simon and colleagues (Simon 1987, Simon *et al.* 1987). To achieve higher-level support, the field may require new design and development approaches, new implementation strategies, and enhanced learning mechanisms mixing OR/MS and AI approaches. These potential needs deserve further research.

A final and general conclusion is that i-DMSS research considered as a whole has not advanced very far in the 1980-2004 time frame. Our results are consistent with reports of the contributions of the AI discipline to the DMSS field (Goul *et al.* 1992, Eom 1998b). According to our analyses, the AI discipline has effectively contributed to the enhancement of some phases of the decision-making process. However, results of this research suggest that these worthy efforts have been isolated and not cumulative. In summary, we believe that our posed framework can be used or extended to evaluate objectively the level of AI components embedded in i-DMSS and the impact of such integration on the decision-making process and outcomes (Mora *et al.* 2005b)

Our strategic review supported empirically by the literature, should not be considered an autopsy of the field. Indeed, the number of papers has grown by about 400% from the first to the second decade. Rather, this research should be considered a motivator that encourages academicians and practitioners in the OR/MS and AI fields to close the identified research gaps. In this way, the i-DMSS community can follow the long-term and strategic research objective set forth by Professor H. A. Simon (Simon 1987) and colleagues in the 1980s: *"… we should aspire to increase the impact of MS/OR by incorporating the AI kit of tools that can be applied to ill-structured, knowledge-rich, non-quantitative decision domains that characterize the work of top management and that characterize the great policy decisions that face our society*".

References

Alavi M, Carlson P (1992) A review of MIS research and disciplinary development. Journal of Management Information Systems 8(4): 45-62

Angehrm A (1997) Computer that critize you: stimulus-based decision support systems. Interfaces 23: 3-1

Baidoo R, White E, Kasper G (2004) Information systems' cumulative research tradition: a review of research activities and outputs using pro forma abstracts. Proceedings of the AMCIS Conference, August 6-8, New York, USA, 4195-4202

Bellman R, Zadeh L (1970) Decision-making in a fuzzy environment. Management Science 17(4): 141-164

Bolloju N, Khalifa M, Turban E (2002) Integrating knowledge management into enterprise environments for the next generation decision support. Decision Support System 33(2): 163-176

Bonczek R, Holsapple C, Whinston A (1982) A generalized decision support systems using predicate calculus and networking data base. Operations Research 29(2): 263-281

Cheng C, Holsapple C, Lee A (1994) The impact of periodicals on expert system research, IEEE Expert 9(6): 7-14.

Cheng, L, Holsapple C (1996) Citation-based journal rankings for AI research: a business perspective. AI Magazine 17(2):87-97.

Cuena J, Molina M (2000) The role of knowledge modeling techniques in software development: a general approach based on a knowledge management tool. International Journal of Human-Computer Studies 53(3): 385-421.

Dhar V, Stein R (1998) Intelligent Decision Support Methods. Prentice-Hall : Upper Saddle River, NJ , pp 7-14

Doyle J, *et al.* (1996) Strategic directions on artificial intelligence. ACM Computing Surveys 28(4): 653-670

Drucker PF (1967) The effective decision. Harvard Business Review 45: 92-8

Elam J, Huber G, Hurt M (1986) An examination of DSS literature (1975-1985), In: Decision Support Systems: a decade in perspective, IFIP WG8.3 Working Conference on Decision Support Systems, The Netherlands, pp 1-17

Elam J, Konsynski B (1987) Using artificial intelligence techniques to enhance the capabilities of model management systems. Decision Sciences 18: 487-501.

Eom S (1988a) An overview of the contributions to the decision support systems area from artificial intelligence. Proceedings of the ICIS Conference, Helsinki, Finland, December 13-16, pp 149-151

Eom S (1998b) The intellectual development and structure of decision support systems: 1991-1995. Omega 26(5): 639-657.

Forgionne GA. (1991) Decision technology systems: a vehicle to consolidate decision making support. Information Processing and Management 27(6): 679-697.

Forgionne GA, Mora M, Cervantes F, Kohli R. (2000) Development of integrated decision making support systems: a practical approach, In: Proceedings of the AMCIS Conference, Long Beach, CA, USA, August 10-13, pp 2132-2134.

Forgionne GA, Mora M, Cervantes F, Gelman O. (2002) I-DMSS: a conceptual architecture for the next generation of decision-making support systems in the Internet Age, In: Proceedings of the DSIAge Conference, Cork, Ireland, July 3-8, pp 154-165

Forgionne GA, Kohli R, Jennings D (2002) An AHP analysis of quality in AI and DSS journals. Omega 30(3): 171-183

Glass R, Ramesh V, Vessey I (2004) An analysis of research in computing discipline. Communications of the ACM 47(6): 89-94

Goul M, Henderson J, Tonge F (1992) The emergence of artificial intelligence as a reference discipline for decision support systems research. Decision Sciences 23: 1263-1276

Gray P, Watson H. (1996) The new DSS: data warehouses, OLAP, MDD and KDD, In: Proceedings of the AMCIS Conference, August 16-18, Phoenix, AZ, USA, 1996.

Jacob V, Moore J, Whinston AB (1998) Artificial intelligence and the management science practitioner: rational choice and artificial intelligence. Interfaces 18(4): 24-35

Holsapple C, Johnson E, Manakyan H, Tanner J (1995) An empirical assessment and categorization of journals relevant to DSS research. Decision Support Systems 14(4): 359-368

Holsapple CW, Whinston AB (1996) Decision Support Systems: A Knowledge-Based Approach. West Publishing : St. Paul, MN.

Howard RA (1968) The foundations of decision analysis. IEEE Transactions on Systems Science and Cybernetics 4: 211-219

Huber GP, McDaniel RR (1986) The decision-making paradigm of organizational design. Management Science 32(5): .572-589

Huber GP (1990) A theory of the effects of advanced information technologies on organizational design, intelligence, and decision making. Academy of Management Review 15(1): 47-71

Liang T (1988) Development of a knowledge-based model management system. Operations Research 36(6): 849-863.

Little J (1986) A research opportunities in the decision and management sciences, Management Science 32(1): 1-13

March S, Smith G (1995) Design and natural science research on information technology. Decision Support Systems 15(4): 251-266

Mentzas G (1994) Towards intelligent organizational information systems. International Transacctions in Operational Research 1(2): 169-187

Morrison J, George J (1995) Exploring the software engineering component in MIS research. Communications of the ACM, 38(7): 80-91

Mora M, Forgionne GA, Gupta JND, Cervantes F, Gelman O (2003) A Framework to assess intelligent decision-making support systems, In: Proceedings of the KES 2003 Conference, Oxford, UK, September 3-5, LNAI 2774, Springer-Verlag, Berlin, pp 59-65.

Mora M, Forgionne G, Gupta JND, Cervantes F, Gelman O (2005a) Intelligent decision-making support systems: a strategic research agenda. Journal of Decision Systems 14(1-2): 179-196

Mora M, Forgionne F, Cervantes F, Garrido L, Gupta JND, Gelman O (2005b) Toward a comprehensive framework for the design and evaluation of intelligent decision-making support systems (i-DMSS). Journal of Decision Systems 14(3):321-344

Manjarres A, Pickin S (2002) Describing generic expertise models as object-oriented analysis patterns: the heuristic multi-attribute decision pattern. Expert Systems 19(3):142-169

Nolan R (1991) The strategic potential of information technology. Financial Executive, July-August : 25-27

Papamichail K, French S (2003) Explaining and justifying the advice of a decision support system: a natural language generation approach. Expert Systems with Applications 24(1): 35-48.

Parnas D (1985) Software aspects of strategic defense systems. ACM SIGSOFT Software Engineering Notes 10(5): 15-23.

Rockart J, Tracy M (1982) The CEO goes on-line. Harvard Business Review 60(1): 81-93.

Rowe AJ, Davis SA (1996) Intelligent Information Systems: Meeting the Challenge of the Knowledge Era., Greenwood Publishing : Westport CT

Sage A (1981) Behavioural and organizational considerations in the design of information systems and process for planning and decision support. IEEE Transactions on Systems, Man and Cybernetics 11(9): 640-678

Satty T (1998) Reflections and projections on creativity in Operations Research and Management Science: a pressing need for a shift in paradigm. Operations Research 46(1): 9-16

Saunders C (2004). Document on line at: http:// www.isworld.org/ csaunders/rankings.htm.

Simon HA (1987) Two heads are better than one: the collaboration between AI and OR. Interfaces 17(4):. 8-15

Simon HA, Dantzing G, Hogart R, Plott C, Raiffa H, Schelling T, Shepsle K, Thaler R, Tversky A, Winter S (1987) Decision making and problem solving. Interfaces 17(5): 11-31

Simon HA (1997) Administrative Behavior: a study of Decision-Making process in Administrative Organizations. Free Press : New York, 4th. Ed.

Sprague R (1980) A Framework for the development of decision support systems. MIS Quarterly 4(4): 1-26

Turban E, Watson H (1994) Integrating expert systems, executive information systems and decision support systems, In: Decision support and executive information systems, Englewood Cliffs, Prentice-Hall, pp 399-407.

Turban E, Aronson J. (1998) Decision Support Systems and Intelligent Systems, Prentice-Hall: Upper Saddle River, 5th. Ed.

Wong, R, Bhattacharyya, S (2002) Task-structure analysis: a modularized approach for modeling knowledge intensive process, In: Proceedings of the 35th. Hawaii International Conference on Systems Science (HICSS-35), Big Island, Hawaii, January 7-10, pp 1-10

24

The Optimization of What?

Andrew M. McCosh and Bertha A. Correa-Pérez

Florida International University, Miami, Florida 33199, USA
e-mails: mccosha@fiu.edu, bertha.correa@fiu.edu

This chapter traces the impact of decision support methods, including those based on Artificial Intelligence concepts, from the beginning, through to the present, and concludes with proposals for the future of the profession. Most of the readers of this book are engaged in the creation of models, systems, data and knowledge bases and methodologies. These are all worthwhile tasks, and some of them are seriously complicated and tricky to do. Our goal in this chapter is to encourage colleagues to move up a gear. Since the start in 1965, members of our profession have solved several thousand problems for organizations. The next job is to tackle more world-class issues. We have the skills and the tools to do this. The executives we work with are more computer aware than they were in the 1960s. We ourselves know more about the need for social acceptance than before. The chapter pencils in the history of the DSS concept from the start, then reviews the problems we are collectively tackling now, before moving on to consider the global scale of the challenges that lie ahead.

24.1 Introduction

The chapter seeks to encourage change in the *Decision Support Systems* field. The profession has made great strides since it began in 1965. The present authors believe that we need to make even greater strides in the coming decade, if we are to continue to be valued by those who deploy our output in managing their organizations. The chapter starts with some history from the earliest days of DSS. We then move to a review or "snapshot" of where we have reached now, about half-way through the first decade of the third millennium. Lastly, and most importantly, we make suggestions on what we should be doing next. The world's problems are vast, and require the highest standard of decision analysis if they are to be tackled effectively and in time. We assume that everyone reading this book will be engaged in some way with creating or deploying models, systems, data and knowledge bases, and methodologies. Our hope is that the book will encourage some to apply their talents to the huge problems discussed in Section 24.3.

24.2 Historical Note

We believe we can state with assurance that the concept of DSS was born as an official discipline within the academic study of business, on Thursday March 4 1965. On that date, Michael Scott-Morton, a doctoral student at Harvard Business School, passed his doctoral thesis proposal oral examination, allowing him to start work on the first DSS. As one of the present authors was sharing an office with him at the time, we are certain of the date. The interrogation was lengthy and extremely thorough. Michael's tentative thesis title, "Using a computer to support decision-making by a manager" was the subject of intense scrutiny. He survived. The thesis title was adapted as he progressed through the work, and the final 1971 book was actually entitled "Management Decision Systems: - Computer-based Support for Decision-making". It is important to understand the "aura of the era". *Operational Research* had had a pretty good Second World War, in which a number of important military logistical achievements could be chalked up to the credit of the concepts of that subject. When "peace broke out in Europe" as the Wall Street Journal so eloquently put it, it was natural for practitioners to turn their skills to civilian applications, and quite a few successes could be claimed there as well. By the end of the 1950s, many refineries were being optimized by linear programs, many pipelines were being managed by simulation models, and some practitioners started to feel, and a few started to claim, that they had the whole management task pretty well under control. Wiser heads among them were much more cautious, notably J. Forrester, who observed in 1961 that "management science has not yet reached a level that would make it effective as a supplement to the skillful practice of top management as an art".

By the start of the 1960s, the decision sciences had had, perhaps without noticing it, a Hegelian moment. The thesis that science could solve the mysteries of managing had been put forward. The antithesis became apparent in the late 1950s, as the vagaries of human emotionality and a repudiation of orthodox rationality applied the brakes to the continuing development of purely scientific approaches to managing. The third step in Hegel's philosophy was the synthesis. The search was on for a compromise, or a new method, that would acknowledge the good features of operational research and other similar approaches as well as coping with the sciences' failure to capture the important experience-based and emotionally grounded aspects of top management behavior. Aristotle has taught us that we never learn how to behave towards others by reading and being taught, we learn these things by practice and regular exposure.

We would contend that the creation of *Decision Support Systems* as a concept was a significant part of this search for a new Hegelian synthesis. In these developmental months in 1964 and 1965, we probably did not realize it. I know that Michael was very pleased to have passed his thesis oral, but I suspect that he did not realize at that time that he had invented a new subject in the process.

It is perhaps worthwhile to note that the project Michael carried out at the Westinghouse factory for the manufacture of "white goods", such as refrigerators and industrial laundry machines, was a group decision support system (there were three executives directly aided by it), an organizational decision support system (the entire production planning process, and production, was dominated by its output), and a knowledge management system (stored data were carefully husbanded for reuse).

In these respects, and in others, the first DSS was a model for the future. A significant portion of the decision process was carried out by the DSS, but another significant portion was done by the three executives on the basis of their years of experience in running the plant. The executives' role was largely a matter of applied intuition, particularly with reference to anticipated movements in the markets for white goods in various states and cities. This book is concerned, in part, with intelligent DSS. When the DSS concept was in its infancy, the assumption was made that the DSS would do the "number-crunching" parts of each problem, while the intelligence would be provided by a human with the required experience and skills. The considerable progress made by *Artificial Intelligence* practitioners over the last forty years has gradually made it possible to reduce the intuitive elements, replacing some of them by AI methods. So far, we have not been able to find any organizational decision support systems, which address important and/or strategic questions, from which the intuitive contributions of executive personnel have been excised. This is partly because the judgment of the executives is likely to be more finely tuned than those of the model builders, and also because the executives do not believe in or trust the models to do the whole job, unless (unusually) they wrote the model themselves.

Throughout the 1970s, the gospel was spreading. Steve Alter did an excellent survey book, showing where we had got to by 1980. In that year, the International Federation for Information Processing [IFIP] began formal operations in DSS, with a series of conferences that eventually led to the formation of a working group on the topic, under the early leadership of Michel Klein, Henk Sol, Leif Methlie, Andrew Whinston, and Enid Mumford, amongst numerous others.

Since then, a very large number of people have become involved in promoting and creating DSS and in strengthening the theory underlying the basic concepts. Still, it seems to be true that most of the basic ideas are still as they were at the start. The fundamental idea of using a combination of an experienced manager using his judgment with the data-storing and model-preserving capabilities of the computer was mentioned on page twelve of Michael's thesis, written about 35 years ago. It is still a feature of almost every DSS in practical use. We can say, therefore, that a huge amount of progress has been made. There are now several thousand people who are engaged in DSS projects of very diverse kinds, and many of them are solving problems for organizations, which we really could not handle at the beginning.

24.3 A Snapshot of the Present

In this next segment, we offer a "snapshot" of where the decision analysis profession has reached now. Sadly, the globally famous *Artificial Intelligence* library at Edinburgh University was destroyed in a fire just before Christmas in 2002. The bulk of their justly famous collection on DSS went literally up in smoke. We therefore rely on the very good collection at Florida International University, which can tell us a lot about the progress we have made.

Any collection of this kind is bound to omit a lot of favorites. We are simply seeking to show that the DSS profession has made great progress and should be ready to tackle some larger-scale problems, to be discussed in the section on our future. Our profession's efforts so far have impinged on (a) the efficiency of organizational operations, (b) the effectiveness of organizational operations, and on (c) the veracity or reliability of the outputs of organizations. Our profession has impacted on (d) the profitability of enterprises, and (e) the agility with which organizations adapt to new situations. In achieving these valuable objectives, the profession has been aided by a transforming growth of DSS understanding, and understanding of computing in general, among the executive ranks. Concurrently, there has been an important improvement in the level of business understanding amongst DSS practitioners, and other computing professionals. In some ways, the most important single change has been [f] a marked advance, over the last forty years, in the willingness of DSS practitioners to understand the vital importance of the belief systems, and social structures, within the organizations they are working with and for. The availability of various kinds of clever software has also helped a lot. *Artificial Intelligence* (AI) systems have played an important role during the last few years. These systems are agents, and they are designed to take part of the burden off a human's shoulders. They are software artifacts, designed to fulfill a limited but still significant role in an organizational process or function. Intelligent agents have been deployed in many industries and many parts of government (including health and education). Several important AI applications have been dedicated to improving the actual process of decision support itself. AI has been evolving and developing since the 1960s, and is now making a real impact in certain companies and other entities.

24.3.1 Efficiency of Operations

The very first DSS was a planning efficiency model, to help the managers of a Westinghouse factory plan their production and distribution of laundry equipment. (Scott-Morton, 1971). The contribution to efficiency was a matter of cutting the total time to create the plan to one day per month instead of 22 days, and freeing up five days of the managers' time. A clear success, this model relied on Herbert Simon's "Intelligence, design, choice" analysis of the decision process.

From the financial field, a DSS was presented, which was claimed, that would do a better job of pricing efficiency for an initial public offering than previous methods had managed (Jain and Nag 1996). Microsoft, for instance, had lost 30% of a capital issue because of inefficient underpricing by their bankers. Neural network methods were used to avoid this leakage.

Wagner and Davis 2001 have devised an algorithm that will help the analyst of a mechanical system trace a fault. Single-item discrete sequential search models are claimed to display "fault-tracing efficiency" in searching for the defect, in that the models find the problem quickly enough to preserve customer goodwill.

There are obviously many other kinds of efficiency, and every business would have to define their own particular version. A significant number of the early DSS models put into use were designed with some variant of operational efficiency as the target to be achieved.

This is still true. Basnet Foulds *et al.* (1996) have just reported a more efficient way of delivering milk, while Tabucanon *et al.* (1995) have shown how a service technician can more efficiently find his way through the labyrinthine traffic jams of Bangkok.

Business operations continue to get more complicated and to move faster with every passing day. The Internet brings new challenges along with its obvious opportunities. Business life grows unremittingly more competitive. Strategies and processes have to keep pace if the entity is to remain in operation and profitable. The use of AI and the emergence of new technological resources have changed the ways in which senior managers work. Many more are moving closer to the major decisions, displacing staff analysts, than was true before.

A system, that helps banks to decide how to improve and enhance their business processes, is “Intelligent Bank Re-engineering System” (Min *et al.* 1996). This intelligent agent helps banks create systems, which will spot problems, and find opportunities, that satisfy the targets, and financial plans to which they aspire.

One example of AI enhancing the effectiveness of a business is in the field of auditing. The EDGAR system was created by the Securities and Exchange Commission to enable access to all the required accounting information produced by businesses quoted on one of the US exchanges (Nelson *et al.* 2000). This is a big help to auditors of the registrant, but also to the auditors of companies who trade with the registrant. The EDGAR-Analyzer (Gerdes 2003) was added to the system more recently, to enhance the efficiency with which the SEC could enable access, as the Freedom of Information Act requires.

Another governmental application of AI is the use of knowledge management in law enforcement. High-speed communications between law enforcement entities can be crucially important in dealing with certain kinds of offence. COPLINK Connect (Chen *et al.* 2002) utilizes a user-friendly interface that integrates multiple data such as incident records, mug shots and gang information. This allows many police departments to share data easily. COPLINK Connect is being deployed at Tucson Police Department in USA.

Legal firms in Australia have been aided by an intelligent agent called “Split-Up” (Zeleznikow and Nolan 2001). This is a prototype system that works out how to distribute the communal property after a divorce. This system was implemented using the current legal rulings and statutes.

Knowledge management, data mining, expert systems, and knowledge acquisition among others, are fields of study that are linked closely to the AI area. The Supplier Cost Reduction Effort (SCORE) system (Walker 1998, Hartley *et al.* 2002) helped Chrysler improve from a $2.6-billion loss in 1995 to a profit of $3.5 billion in 1996. SCORE, the Chrysler’s collaborative computer-support system has evolved along with technology and its current version contains AI technology and its database includes not only product information, but also user profiles and suppliers’ and customers’ information. SCORE enhanced the relationship between Chrysler and suppliers while improving the quality in services as well as in products.

Virtual reality is another field that involves the implementation of AI systems. Currently, the use of cellular phones, PDA-palms and Handhelds to type information and connect to the internet, although it might be a difficult task, has become very common in the business field. Virtual Keyboard (VKB), an Israeli IT company, developed a device that includes a system that uses a laser ray to project a virtual keyboard onto any plane surface. The first virtual keyboard was presented in 2000 at Comdex expo show in Las Vegas. Although there are still some doubts about the accuracy of the system, Siemens agreed to promote the product in the European market, and Siemens Procurement & Logistic Services became the exclusive distributor of the virtual keyboard in Germany. Some new prototypes are being developed by other companies; for instance, Samsung presented the "Scurry" prototype, a bracelet that detects the movements of the hand and fingers, which also avoids the need of an actual keyboard.

24.3.2 Impact on the Effectiveness of Organizations

The second dimension of concern in the study of DSS as they currently affect organizations relates to effectiveness. Do these systems get the organizational mission(s) accomplished, or at least help towards them? Once again the DSS profession can make some significant claims. Grabot *et al.* (2004) have devised a management algorithm that copes with missing personnel, machine failures, and market shifts, showing how to get the task done despite such obstacles. Dowling and Louis (2000) have examined the effectiveness of meetings in bureaucratic entities. They have shown that a traditional meeting with everyone in the same boardroom at the same time is a horribly expensive way to behave. Instead, they advise the use of computer-assisted asynchronous meetings. They are a lot cheaper, they involve less travel, they consume fewer person-hours, and they generate more and better ideas. Our department chairman was interested in this idea, and suggested we should hold a meeting to discuss it. Clearly, we have a way to go yet!

In the hotel industry, the name of the game is revenue optimization. Baker *et al.* (2004) have demonstrated that the model in general use is invalid. Previous research and much practice had assumed that the demand for a service package was independent of which service packages were available for sale. They claim this is wrong, that there is dependence, and it can be used to boost revenues by 16%. This is a significant amount in a high fixed cost business.

In the railway system, safety is of over-riding importance. Brezillon and Pomerol (2004) have shown how to handle the infinite complexity of the safety riddle. They have shown how to trim the decision tree to leave only a few major branches, and then prepare scenarios and reactions to them. Safety is a vital issue in the management of oil pipelines too. Larichev *et al.* (1996) have offered a very powerful discussion of a major decision in the far North of Russia, to develop a gas field safely in an area of serious pollution risk. Still on the subject of safety, three more colleagues have studied groups of medical diagnostic models, and have shown that using a series of diagnostic decision support systems works better than using any one of them. Mangiameli *et al.* (2004).

In large manufacturing companies, an appropriate decision in the production plan might be crucial. Yang and Mou (1993) have explained how Dalian Dyestuff plant, one of the largest chemical plants in China, solved a manufacturing decision decentralization crisis by implementing an intelligent DSS. This includes two expert systems that are based on the knowledge collected from experienced managers and are able to generate a proposed production plan that can be improved by the knowledge engineer. The intelligent DSS enhanced the manufacturing decision-making process, improved the customers' service quality, and increased the company's profit by over $1 million each year.

24.3.3 Veracity and Reliability

An important development of the last twenty years or so is the growth and improvement of the procedures for verification and validation of the results of DSS and other computer-dependent modeling procedures. The valuable work of Amos Tversky has been seized upon by several workers in the DSS area. The idea is to establish whether judgments made by people can be relied upon in arriving at collective or corporate judgments on important business or organizational questions.

Lai (2001) has, for instance, demonstrated that when people are asked to make a series of judgments of the form "Is X equal to Y?", they would generally be fairly consistent in their answers. However, if they were asked to say whether X was half of Y or X was a third of Y, they were seriously adrift. This insight is valuable in ensuring that we ask the correct questions so as to get valid opinion measures.

Oleg Larichev and colleagues have been working on multicriteria decision making for more than forty years. They have recently published a new method of decision support for these difficult problems. The method involves sorting the various possible options into ordered decision classes. Like the tree-pruning method of Brezillon and Pomerol, above, the Larichev method greatly reduces the number of items that have to be considered.

In the banking industry, deciding whether a loan should be approved might be a complex and delicate task for a bank officer. Langley and Simon (1995) presented the applications of machine learning and rule induction implemented at American Express UK. The loan officers used an automated statistical method to help make decisions regarding loan approvals. The method generated decisions for 85% to 90% of the loan applications. The other 10% to 15% were done manually, but these were only about 50% accurate. American Express developed an intelligent system to overcome this problem. The intelligent system based on machine learning and rule inductions increased to 70% the accuracy of the answers given and also provided an explanation of the decision made. The knowledge acquisition was based on examining 1014 loan applications.

t is sometimes difficult to tell from the text of an article just how secure the results would be.

Jimenez *et al.* (2003) describe a DSS based on an additive or multiplicative multiattribute utility model for identifying optimal strategy. Their system admits imprecise assignments for weights and utilities and uncertainty of the multi-attribute strategies, which can be defined in terms of ranges for each attribute instead of single values. They include an application, that deals with the important problem of the restoration of a contaminated lake. The ability to adjust the variables is obviously a valuable feature of this DSS as of many others. The main advantage of this kind of model is that you can then do sensitivity tests to see which of them are critical.

A different kind of veracity measurement comes into play when we are dealing with Internet transactions. There have been quite a number of fraudulent transactions reported in the newspapers, but it is hard to tell whether these are genuine. Ba *et al.* (2003) claimed that the use of a trusted third party in the online deal can solve security problems. Certainly the banking industry has attempted to fill this gap in the electronic market place. By taking a credit card number from the customer and a corporate product/customer code from the supplier, the banks replace both with a new bank-generated number to make the transaction. The credit card number remains safe. Or at least it is as safe as the bank.

Yet another variety of veracity problem relates to intruders into networks. Zhu *et al.* (2001) have studied intrusion-detection systems. These help administrators to prepare for and deal with network attacks. These systems gather data from many sources and use these to predict misuse of the system. Their paper studies three data-mining systems, and they conclude that the group of data-mining systems they have named "Rough sets" provide the highest level of accuracy and level of assurance.

When evaluating veracity and reliability, some other important fields of study include the applications of *Artificial Intelligence* in medicine and education. Indeed one of the very first nonmechanical applications of the subject, MYCIN, was in medical diagnostic work (McCarthy 1984). The MYCIN expert system was developed at Stanford University to perform blood-infections analysis and recommend an appropriate treatment for each patient.

Much more recently, the ability of an AI learning technique to work out what a patient is likely to need has been described (Liu Sheng *et al.* 2000). A radiological case library is employed in conjunction with a patient-specific knowledge-acquisition procedure. This allows prior knowledge, filed by other radiologists, to be studied as well as data from the direct examination. The learning tool in the system uses these data to build the patient's image automatically, without the participation of knowledge engineers and radiologists. This can reduce time and resource requirements for constructing and consequently updating the patient's image-retrieval knowledge base.

A knowledge-discovery approach is found in Kusiak *et al.* (2003). The research applies data mining for predicting survival for kidney dialysis patients. The data gathering was done at four different locations of The University of Iowa Hospitals and Clinics. After applying data transformation, the results indicated a vast potential for making accurate decisions for individual patients, as well as for patient classification. Further research will be performed.

In education, a recent development has been an AI application, that teaches teachers how to assess an essay. Zeleznikow and Nolan (2001) designed and developed IFDSSEA, an intelligent fuzzy decision support system that assists the instructors in the evaluation of the student's essays. In addition, IFDSSEA aids inexperienced instructors to develop their essay-assessment skills. It also improves the teacher's ability to use district-wide scoring rules. IFDSSEA has been implemented in two schools in New York City School District Six.

24.3.4 Profitability

It would almost certainly be true to report that most of the earlier DSS models were prepared on quantitative problems. Issues of profitability were familiar to financial people, to accountants, to operational researchers, and other numerical people. In addition, there were lots of numbers available for them to experiment with. It is simply harder for the people working at the softer end of the management scale to get going, while the financial workers had most of the numbers they needed in the Wall Street Journal, the Financial Times, and a smallish number of other publications. It did not take long for the numbers in these papers to become available in databases, which made DSS projects in the financial area even easier to develop, while the people interested in modeling the qualitative elements of the management problem were struggling to make any sense of the task of putting a meaningful number on any of the subtle dimensions they wanted to study. For these reasons and others, there was a substantial early flow of papers on financial issues and the decisions that these involved. Product profitability was addressed by McCosh and Scott-Morton (1968). Some papers on mergers and acquisitions were provided by McCosh *et al.* (1969). There were many others, dealing with a multitude of financial issues, including share trading, option studies, bond operations, and corporate fund raising. By 1990, Trippi and Turban had enough material to assemble an entire book on investment-management models.

One of the most interesting papers on profitability in relatively recent years has been the paper by Benbasat and Todd (1996), which considers the cognitive cost-benefit sum in terms of how a model would be built. The amount of time and effort it would take to build it will be compared with the benefits it may bring. The cost of building a model and the cost of using the model will both influence the amount of effort put in by the model builder and by the model-funder. This calculation is not trivial.

Over the years there has been a spate of papers attempting to predict bankruptcy. Beaver (1966) and Altman (1968) led the way, and a platoon of other authors followed. In 1996 Serrano-Cinca produced a neural net self-organizing model for predicting solvency of companies. Fairly recently, Castarella *et al.* (2000) have produced a variable that tries to behave as a proxy for the auditors opinion about the solvency of a client firm. The results seem to be rather uncertain. This literature has become less fashionable in recent years, partly because predictions of impending bankruptcy tend to become self-fulfilling, if the client firm's banker happens to read the paper.

Considerable interest still exists in the management of financial institutions. Moynihan *et al.* (2004) have produced a model for handling an institution's exposure to interest rate risk.

24.3.5 Organizational Agility

There have been many sagas about large, old, cumbersome companies, that have failed to adapt to changing circumstances. Some of them have been quite young companies, but they failed to notice that they were in danger until too late. The current (2004) state of many of the world's major airlines might serve as an example. It is pleasing, therefore, to be able to list a couple of success stories. Adam and Pomerol (1998) have shown how a massive monopolist could display remarkable agility when threatened by the competition commission in Brussels. The board decided information technology was a crucial element in their future survival, and focused resources so as to alter the status and influence of IT as a strategic focus of the business.

James and Thompson (2000) have studied the Housing and Development Board (HDB) of Singapore. The HDB has the task of providing affordable, high-quality public housing to Singaporean citizens. The board is working to simplify the data set and the process of gaining access to it. The process was previously very tedious, because too many data were stored. A new government minister has prodded the bureaucrats so that they are now much more responsive to citizens' demands for housing information.

Recent developments in machine learning allow the software to take a more active role. If the DSS has sufficiently well-defined interactive capabilities, it may be able to generate valid and robust computational answers even when the information is incomplete. If the interaction is typical, it will involve negotiation and the resolution of conflicts. A superior form of software may be needed. One such might be PERSUADER, created in 1993 by Sycara Katia. This learning machine is designed to help resolve conflicts. It uses both case-based reasoning and *Artificial Intelligence* methods to learn from the cases presented to it, and also to learn from previous failures so as to avoid repeating them.

The typical big company generates huge heaps of data. These can be stored in data warehouses and similar devices, and the data must be cleansed and updated periodically. The data stored in this way may well help decision makers with their task. (Nemati *et al.* 2002). A lot of the crucial information, however, is not stored in this way. A lot of the most important material walks out of the office each night in the brains of staff members. Unless the firm takes steps to capture some of this information, it may be lost for good, if the employee changes jobs. Even if he changes jobs within the company, the loss may be acute. The new generation of knowledge warehouse should not only facilitate capturing and coding of knowledge, but also improving the retrieval and sharing of knowledge across organizations. These organizational processes may be automated by using one or more of a range of software devices that facilitate information management.

Venkataramani (1997) developed SoftCord, an intelligent agent for coordination of software development projects. SoftCord has been used by a "Big Six" consulting firm to help create a real estate management system for a communications technology firm on the Fortune 50 list. The system manages the property owned and leased in North America by the communication technology firm. SoftCord can record new problem-solving ideas and methods in a knowledge database. It can also detect whether and where a conflict among two or more tasks has arisen or seems likely to. SoftCord is an autonomous program that is able to respond to queries from individual designers. It can also set up and support negotiation sessions between designers to resolve a conflict, such as inconsistencies in the design. Venkataramani Johar mentions the possibility of integrating SoftCord with gIBIS, by Conklin and Begeman (1987), a hypertext system designed to capture early design ideas, using color and a high-speed relational database server to ease the task of building and browsing networks, in a manner more in keeping with how designers usually work.

24.3.6 Social Systems and Belief Systems

The most basic belief system that one observes widely in the computer field is positivist. You decide where you want to get to, you consult the relevant workers and managers to make sure you have got the flowchart right, and then you write the first draft of the system. This approach works at times. Berztiss (1996) has shown that this works in certain kinds of business environments, for certain kinds of problem. It does not work very often. One of the present authors has been writing decision models for forty years, and has managed to get a working system that was acceptable to the executive team, and to the relevant workforce, on first presentation, exactly once. The norm is very different. Iivari (1991) has studied the literature of information systems, and has found the positivist tendency to be very influential. He has, however, noted that the DSS literature is less prone to this mechanistic design method.

A solution to this problem is to be found in Carlsson and Leidner (2004). This involves broadening the analytical procedures, and making the design task into a repetitive set of procedures, following each other round a series of loops until the hidden social issues, which positivist systems designers tend to underestimate or even to ignore, have been brought to the surface and dealt with in a manner that is both business friendly and also behaviorally sound. They refer to Layder (1994) who has instructed designers to pay attention to the basic concepts and “knowns” of social science in designing systems and the anticipated reactions to them, but also to pay close attention to the unique mixture of talents, prejudices, traditions, and beliefs held by the organization’s people, of all relevant ranks. Consider the context, then the setting, then situated activity, then the self. How will the affected parties react to the plan? It is recommended that several social measurement methods should be brought to bear on the issue, to see if they broadly agree.

The organization will have a culture. They cannot avoid having a culture, unless the entity is in its first year or two of existence. They may be very cautious and unadventurous. They may be piratical. It is even possible for a firm to be both. Enron, for instance, was quite traditional in managing its pipeline operations, but seriously piratical in managing its finances. Ghosh and Ray (1997) have conducted an experiment, that shows that there is a measurable attribute called “attitude to risk”, and another measurable attribute called “tolerance for ambiguity”. Business people have a fairly high tolerance for ambiguity and are willing to live with quite substantial levels of risk. The more ambiguity there is, the higher they estimate the risks to be, even if the reality is that the risk is just the same as it was before the level of ambiguity was augmented. Given that they are relatively relaxed about the level of risk they are accepting, it is not surprising that they are very sensitive about the competence of their information systems and especially their ability to predict future difficulties. A system designer may be misguided if he spends time on improving the accuracy of a particular system, when the executive would much rather obtain rough information about the impending problem coupled with a fairly precise estimate of when it is likely to happen. This difference in approach may be absolutely central to the level of acceptance a system attains.

24.4 Robotics Protecting and Aiding People

One of the biggest challenges of *Artificial Intelligence* has been replicating the complexity of human motions and behavior. Robotics currently represents an $8 billion dollar industry globally, mainly for welding, painting, and assembly line tasks. The Honda Corporation has ambitions going far beyond such jobs. Their Research & Development Wako Fundamental Technical Research Center in Japan has created ASIMO (*Advanced Step in Innovative Mobility*) a word derived from the Japanese word "ashi" meaning leg or foot and "mo", which stands for movement.

ASIMO is a humanoid robot that took over 18 years of scientific assessment, research, and trial and error experiences before the scientists from diverse disciplines and engineers at Honda accomplished the goal of creating an advanced humanoid robot. Currently, ASIMO performs tasks as a tourguide in museums and as a greeter at high-tech companies in Japan. Furthermore, ASIMO is also used in education performing demonstrations of ASIMO's technology circuits. ASIMO's main purpose is to be a helper to people in need, such as people who lack mobility. It might also be used to assist fighting fire, and removing toxic waste, and other dangerous jobs. One was recently deployed in the Iraq war, to enable a suicide bomber who changed his mind to cut his way out of his bomb suit without endangering the marines nearby.

NASA makes use of robots to perform different missions. For instance, Robonaut is a humanoid robot developed to be able to walk outside the space station. Da Vinci is a robot element of a surgical system used by surgeons to perform complex operations with tiny surgical incisions. Da Vinci has been used in several prostate cancer procedures with successful results. Furthermore, the EndoVia surgical robot is also used by surgeons to perform operations that would have involved large incisions under traditional procedure. Only a minimal incision is needed when the robot is part of the procedure. Moreover, the risk of infection is minimized by the use of robots.

24.5 Security

A relatively new area of concern, which seems likely to get worse before it gets better, is the security of our systems. The physical computers are (generally) well guarded in various ways. The data entering them is less safe. The bulk of the customer population has not yet realized just how important personal data security is becoming.

The only amateurs who really appreciate the problem are the ones who have suffered loss. We do not really have much in print in the DSS literature on security at the moment. The dangers of identity theft are becoming more and more apparent. The possibility of losing control of our social security number is now frightening everyone in America. Losing control of our credit card number is more of a nuisance than a disaster, but it can take a lot of time to straighten everything out.

Ba *et al.* (2003) have put forward the idea of a trusted intermediary. This procedure was mentioned above. The trusted entity takes your credit card number and the company's reference number and issues a third number to both parties to enable the transaction to be enacted. Our bank in Florida is providing this service, but our bank in Scotland is not, as yet. The sooner the customer population catches on, the better.

Perhaps the internal organization of our computer and IT resources can have an impact on safety and security. Ranganathanan and Sethi (2002) assert that firms with centralized IT organizations are less likely to have good procedures for sharing domain knowledge. But good sharing procedures are more likely to make IT choices, including IT security choices, sound. If departments become seriously frightened about security issues, and especially if they are penalized from above if there is a breach, this is likely to make departments clam up even more than before. Such a change in attitude may make cooperative development of advanced systems almost impossible. The situation is hazardous but not hopeless. Zhu *et al.* (2001) show how we can use data mining ideas to spot unauthorized people getting into our machinery. They test several methods, and make recommendations for which to deploy.

24.6 Summary About the Present State of the Field

In summary about the present, it seems clear to us that the DSS profession has a lot to be proud of, and has made major positive contributions to the organizations, that have seen fit to deploy our collective talents along with the AI technology. At this point, we must move to the challenge of the future. This is the core of our chapter. We believe that the profession has slowed down. The first fifteen years saw enormous and rapid progress in terms of results and of methodologies, but in the second fifteen years we seem to have been repeating the considerable triumphs of the past without adding very much that is new. In the second fifteen years, there has also be a divisive tendency, in which more and more organizations have been formed, all of them handling the same kinds of problems, but none of them having a resource base large enough to tackle any of the major global issues. There is a need to be less timid, more forceful, and to tackle larger problems. Not just problems affecting a single entity, but problems, that affect the whole world. Unfortunately, there are quite a few of these.

24.7 Stepping Forward from Now Until 2020

We do not yet know how decision support systems and *Artificial Intelligence* systems will evolve over the next 10 to 15 years. Technology has evolved quickly and this will continue. Decision makers will have to become familiar with AI concepts. Knowing how these systems work and what they can do will help managers enhance product and service quality. In turn, these improvements will lead to faster, more productive, and more complete solutions. The focus of AI systems and DSS is on decision-making efficiency; to let managers spend their time on choosing **what** to focus on. For instance, the Danish Environmental Assessment Institute (DEAI), with the help and support of the Economist, developed a list of major global problems. We were relieved to see that the economists did not expect to be able to solve them on their own. Systems theorists, and still more IT practitioners, have a great deal to contribute to the solution of these colossal problems. We list them, and then discuss a few of them, necessarily in a cursory manner. We hope that some of you may find them challenging and worthwhile. The sponsors, the DEAI, have commissioned major reports by distinguished authors on nine topics (see Table 24.1). They plan to make recommendations to the world's policymakers as to how these problems might be solved. They also, and this may be more important, plan to specify which of the problems should receive priority treatment. The list of problems, in strictly alphabetical order, together with the names of the people who have been asked to write a position paper to inform the debate and their respective home organizations, are:

Table 24.1. List of DEAI's researchers in very hard-difficult decision-making problems.

TOPIC	AUTHOR	LOCATED AT
Armed Conflict	Paul Collier	Oxford
Climate Change	William Cline	Global Development Center
Communicable Diseases	Anne Mills	London School Tropical Med
Education	Lant Pritchett	Harvard School of Government
Financial Instability	B. Eichengreen	Berkeley University of California
Governance+Corruption	S. Rose-Ackerman	Yale
Malnutrition, Hunger	Jere Behrman	Pennsylvania
Population, Migration	Philip Martin	Davis University of California
Sanitation, Water	M. Haneman	Berkeley University of California
Subsidies, Trade Barriers	Kym Anderson	Adelaide

The DEAI is going to get two other experts in the various topics to write a commentary on the first paper. The whole bundle is then going to be passed to a group of nine economists, who are supposed to come up with recommendations that will be conveyed to governments for possible action. The DEAI Unit is quite realistic about its chances of getting anything to happen. The Economist (2004) admits that the title of the project, "The Copenhagen Consensus" is itself optimistic. However, getting the nine economists to agree may turn out to be easier than they expect, because all but two of them are Americans. The Swiss and the Chinaman may well add diversity to the thinking, but they can be outvoted. There can be no doubt that the topics chosen are seriously important.

AI is a key that might be useful in developing some of the solutions. Robotics, for instance, in the persona of ASIMO, already plays a role in **education**. Intelligent (AI) agents can learn from experience and can apply their knowledge to new situations. They can handle difficult tasks, sometimes in dangerous or even toxic situations. Robots can also enhance the skills of humans, as (for instance) the Da Vinci and the EndoVia robots that can perform parts of surgical operations. Educational standards affect the competences of business workforces, and these standards are controlled, or at least heavily influenced, by government. Most of those who will read this chapter are products of the very efficient educational machines of the advanced economies, and may therefore be well equipped to help formulate these choices. We need to spend time thinking about how we can best help the politicians, the educationalists, and the citizenry at large decide on the goals of education. Do we need more professors? Do we need more plumbers? Do we need more carpenters, or just better ones, or both? There seems to be some evidence that the health-care professions are not attracting enough new recruits in many lands. If so, how do we formulate the long-term decisions to make the changes we need? There is a definite need for people in both the decision support systems field and the AI field to help the political decision makers to focus on what is needed, instead of what is fashionable.

Subsidies affect the business subsidized as well as its unsubsidized competitors, and governments create them. There was a case before the World Trade Organization at the time of writing this in which the Americans are accused by the Europeans of subsidizing Boeing while the Europeans are accused by the Americans of subsidizing Airbus. If these are sustained, the real losers will have been the aircraft manufacturers (in other countries) who were put out of business by the subsidies. These real losers will not be regenerated. Whatever modest market share they had is lost to them forever. Perhaps we, as decision analysts, can address the question of exactly why people want to prohibit subsidization. If we consider industries like textiles or sugar, the problem of unfairness becomes much clearer. A rich country can subsidize its own textile producers, or sugar growers, to an extent which a poor country could not. These subsidies would be seriously tough on small countries or poor countries that cannot gain access to the rich markets, possibly to the extent of being insurmountable barriers. The rich countries have all the power in this situation. It is a question of what their ethical posture is concerning trade. Just how much do they want to oppress the poor countries? Just how much do they want to promote and encourage the economies of the poor countries? Subsidies are the mechanisms through which these questions are answered.

AI technology can be used to predict the moves of competitors, consumers, and combatants in situations which may involve **armed conflict**. AI can help in predicting human and business behaviors. For instance, Genoa II is an ongoing AI project at the Defense Advanced Research Projects Agency (DARPA). Its main job is to help create the information technology to help groups of intelligence analysts and operations and policy personnel in their efforts to foresee and frustrate terrorist threats to the USA. The underlying concepts will almost certainly be transferable to other competitive situations, such as business

The amount of energy we consume may be an important determinant of the way our **climate is changing**. We have been heavily influenced by the work of the scientists of the University of Alaska. George Divoky, an ornithologist, has studied climate change in northernmost Alaska, as he watched the guillemots come to their nesting places later and later in each summer over thirty years of observation. He is quantifying climate change in a frightening way. Nobody else has published the year-by-year changes over such a long time at such latitude, 73 degrees north, in his case, on Cooper Island. The icecap is melting, and not slowly. The ice melts in northern Alaska about one day earlier every two years, on average. Our prosperous economies consume energy and dissipate heat as a byproduct. While it is self-evident that we cannot go on doing this for ever, it is also clear that we are not yet sufficiently frightened by the problem to cause our politicians to do something. We, as decision analysts, should be producing solutions, or programs that might lead to solutions. The task of coping with these potential disasters is going to involve decision analysis of the highest order.

In America, we have recently been conducting a nationwide seminar on **governance and corruption**. As a professor of ethics, one of the authors' tranquil and academic little world has been suddenly and seriously enlarged by Enron and Worldcom, Parmalat and Global Crossing, Robert Maxwell and HealthSouth. The frantic and (in our opinion) doomed attempt by the lawyers to enforce morality by law has certainly endorsed the inclusion of that topic in the list. Some of the questions that arise in this area have been with us for centuries. The South Sea Bubble bankrupted many leading Britons just less than three hundred years ago. There have been frauds and fiddles on and off ever since. And even before that, if Martin Luther is to be believed. The priests used to sell indulgences, which allowed the payer to commit assorted offences up to the value on the price list.

The present authors do not claim any expertise about communicable disease, population and migration, malnutrition, and the other topics. We are confident that colleagues will be able to make valued contributions. Formulating the decisions that have to be taken is a huge undertaking in its own right, and doing this well will be a big step towards arriving at the answers. We, as decision scientists, can prosper by tackling these matters. We are very good at analyzing a decision. We are very good at collecting data, that bears on a decision. We have developed a considerable armory of methods for examining opinions. We know how to assess preferences. These are all inexact sciences, but we are at least as good as anyone else at making them practically operable. We have members in nearly every country, all of whom have similar talents. We know, better than anyone else, that there are aspects of every important decision that can be handled by the technology, while there are other aspects, that cannot. The latter group of issues must be handled by executive intuition, an element about which we are more knowledgeable than any other group of people on the planet. We know, collectively, how to extract and refine executive opinion so that it becomes a useable force to be exerted on the problem. We know how to gather opinions from groups of managers, groups of citizens, even groups of animals. We strongly believe we should be shouting a little louder. We can help to attack these big problems. We can help, not only to attack them, but also to resolve them. Because of previous exposure to the field, the authors hope to tackle financial instability, poor governance, education and rampant corruption. Perhaps these are not the most important issues, but they are the root causes of some of the rest.

The goal of this chapter is, we hope, now clear. We want to work towards a world in which the big decisions on the allocation of our resources, which are very limited, are taken by morally strong people, who will also have the moral strength to blow whistles when whistles need to be blown. We want to work to help the policy makers formulate the decisions on resource allocation, by deploying the skills we all have, skills in decision support, skills in decision analysis, skills in option formulation, skills in information technology. We have (collectively) the skills needed to make some of them tractable. We hope some of you may feel like participating in some part of this admittedly hefty assignment.

References

Adam F, Pomerol JC (1998) Context sensitive Decision Analysis based on the investigation of organisational information networks, Context Sensitive Decision Support Systems, Berkeley, Dina, Widmeyer, George, Brezillon, Patrick and Rajkovic, Vlado (Eds.), Chapman Hall, London, U.K., pp 122-145

Altman EI (1968) Financial ratios, discriminate analysis and the prediction of corporate bankruptcy. Journal of Finance 23: 589-609

ASIMO. http://asimo.honda.com

Ba S, Whinston AB, Zhang H (2003) Building trust in online auction markets through an economic incentive mechanism. Decision Support Systems, 35: 273-286.

Baker T, Murthy N, Vaidyanthan J (2004) Service package switching in hotel revenue management systems. Decision Sciences, 33(1): 109-132

Basnet C, Foulds L, Igbaria M (1996) Fleetmanager: a microcomputer based DSS for vehicle routing. Decision Support Systems, 16 : 195-207

Beaver W (1996) Financial ratios as predictors of failure. Journal of Accounting Research. Supp: 71-111

Benbasat I, Todd P (1996) Effects of decision support and task contingencies on model formulation; a cognitive perspective. of Decision Support Systems 17: 241-252.

Berztiss A (1996) A software process for management software systems, Implementing Systems for Supporting Management Decisions, Humphreys, Patrick, Bannon, Liam, McCosh, Andrew M., Migliarese, Piero and Pomerol, Jean Charles (Eds.), Chapman & Hall, London, U.K. pp 21-33.

Brezillon P, Pomerol JC (2004). Using contextual information in decision-making, Context Sensitive Decision Support Systems, Berkeley, Dina, Widmeyer, George, Brezillon, Patrick and Rajkovic, Vlado (Eds.), pp 158-173.

Carlsson S, Leidner DE (2004) Contextual design of management support systems. In, Context Sensitive Decision Support Systems, Berkeley, Dina, Widmeyer, George, Brezillon, Patrick and Rajkovics, Vlado (Eds.), Chapman & Hall, London, U.K. pp 88-105

Castarella JR, Lewis BL, Walker PL (2000) Modeling the audit opinions issued to bankrupt companies: a two-stage empirical analysis. Decision Sciences, 31(2): 507-530.

Chen H, Schroeder J, Hauck RV, Ridgeway L, Atabakhsh H, Gupta H, Boarman C, Rasmussen K, Clements AW (2003) COPLINK Connect: information and knowledge management for law enforcement. Decision Support Systems 34 (3) : 271-285

Conklin J, Begeman ML (1997) gIBIS: a hypertext tool for team design deliberation, Chapel Hill, NC, U.S.A. pp 247-251

Da Vinci. http:// http://www.davinciprostatectomy.com

Dowling KL, St. Louis RD (2000). Asynchronous implementation of the nominal group technique: is it effective? Decision Support Systems 29 (3): 229-248

EndoVia. http://www.mos.org/cst/article/5403/

Forrester JW (1961). Industrial Dynamics, Productivity Press. Cambridge, MA. U.S.A. pp 464

Gerdes J (2003) EDGAR-Analyser: automating the analysis of corporate data contained in the SEC's EDGAR database. Decision Support Systems 35: 7-29

Ghosh D, Ray MR (1997) Risk, Ambiguity, and decision choice; some additional evidence. Decision Sciences 28(1): 81-91

Grabot B, Blanc JC, Binda C (2004) A decision support system for production activity control. Decision Support Systems 16: 87-101

Hartley JL, Greer BM, S Park (2002) Chrysler leverages its suppliers' improvement suggestions. Interfaces 32(4): 20-27

Kusiak A, Shah S, Dixon D (2003) Data mining based decision-making approach for predicting survival kidney dialysis patients. In DD Feng and ER Carson (Eds.), Modeling and Control on Biomedical Systems 2003, Proceedings of the IFAC 2003 Symposium on Modeling and Control of Biomedical Systems, Melbourne, Australia, Elsevier; Amsterdam, The Netherlands, August 2003, pp. 35-39

Iivari, J (1991) A paradigmatic analysis of contemporary schools of IS development. European Journal of Information Systems 1(4): 249-272

Jain BA, Nag BN (1996) Artificial Neural network models for pricing initial public offerings. Decision Sciences 26(3): 283-302

James A, Thompson SH (2000) Management issues in data warehousing: insights from the housing and development board. Decision Support Systems 29(1): 11-20.

Jimenez A, Sixto R, Mateos A (2003) Decision support system for multiattribute utility evaluation based on imprecise assignments. Decision Support Systems, 36 (1): 65-79.

Lai SK (2001) An empirical study of equivalence judgements vs ratio judgements in decision analysis. Decision Sciences, 32(2): 277-302

Langley P, Simon HA (1995) Applications of machine learning and rule induction. Communications of the ACM, 38(11): 55-64

Larichev OI, Andreyeva EN, Sternin MY (1996) System for preparing management decisions: - a gas pipeline siting study. In, Implementing Systems for Supporting Management Decisions, Humphreys, Patrick, Bannon, Liam, McCosh, Andrew M., Migliarese, Piero and Pomerol, Jean Charles (Eds.) Chapman Hall, 1, London, U.K. pp 249-260

Layder D (1994) Understanding Social Theory, Sage: London.

Liu S, Olivia R, Wei CP, Hu PJ, Chang N (2000). Automated learning of patient image retrieval knowledge: neural networks versus inductive decision trees. Decision Support Systems 30 (2): 105-124.

Mangiameli P, West D, Rampal R (2004). Model selection for medical diagnosis decision support systems. Decision Support Systems 36(3): 247-259

McCarthy J (1984) Some Expert Systems need common sense. Stanford University 1984. (www.stanford.edu)

McCosh AM, Scott Morton MS (1968) Terminal costing for better decisions. Harvard Business Review 46(3): 22-38.

McCosh AM, Hawkins DF, Lampe JC (1969) Time-shared mergers and acquisitions. Mergers and Acquisitions 4(1): 31-42.

Min D, Moonkee K, Jong R, Kim WC, Min D, Ku S (1996) IBRS: Intelligent bank reengineering system. Decision Support Systems 18 (1): 97-105

Moynihan GP, Purushothaman P, McLeod RW, Nichols WG (2004). DSSALM: A DSS for asset and liability management. Decision Support Systems 33: 23-38

Nelson KM, Kogan A, Srivatava RP, Vasarhelyi MA, Lu H (2000) Virtual auditing agents: the EDGAR agent challenge. Decision Support Systems 28(3): 241-253

Nemati HR, Steiger DM, Iyer LS, Herschel RT (2002) Knowledge warehouse: an architectural integration of knowledge management, decision support, Artificial Intelligence and data warehousing. Decision Support Systems 33(2): 143-161

Ranganathan C, Sethi V (2002) Rationality in strategic information technology decisions: impactg of shared domain knowledge and IT unit structure. Decision Sciences, 33(1): 59-86

Serrano-Cinca C (1996) Self organising neural networks for financial diagnosis. Decision Support Systems 17: 227-238

Scott-Morton MS (1971) Management Decision Systems: - Computer-based Support for Decision-making. Harvard Business School, Boston, MA, USA, pp 1-163

Sycara, KP (1993) Machine learning for intelligent support of conflict resolution. Decision Support Systems 10 (2): 121-136

Tabucanon MT, Kovavisaruch L, Malaivongs K (1995) Microcomputer based heuristic approach to vehicle routing for after-sales servicing. Decision Support Systems 13: 195-205

The Economist. March 6, 2004, pp. 68.

Trippi RR, Turban E (1990) Auto learning approaches for building expert systems. Computers and Operations Research 17: 553-560

Venkataramani J (1997) SoftCord: an intelligent agent for coordination in software development projects. Decision Support Systems 20 (1): 65-81.

Walker D (1998) Supply chain collaboration saves Chrysler $2.5 billion and counting. Automatic I.D News, (August) pp. 60.

Wagner BJ, Davis DJ (2001) Discrete sequential search with group activities. Decision Sciences 32(4): pp. 557-573

Yang D, Mou W (1993) An integrated decision support system in a Chinese chemical plant. Interfaces 23(6): 93-100

Zeleznikow J, Nolan JR (2001) Using soft computing to build real world intelligent decision support systems in uncertain domains. Decision Support Systems 31(2): 263-285

Zhu D, Premkumar G, Zhang X, Chu, CH (2001) Data Mining for Network Intrusion Detection. A comparison of alternative methods. Decision Sciences, 32(4): 635-6

Contributor Biographies

Frédéric Adam is a Senior Lecturer in the Business Information Systems Group at University College Cork in Ireland. He holds PhDs from the National University of Ireland and Université Paris VI (France). His research has been published in international journals including Journal of Strategic Information Systems, Decision Support Systems, and Journal of Information Technology. He is the Editor-in-Chief of the Journal of Decision Systems and the co-author of "The Entreprise Resource Planning Decade: Lessons learnt and Issues for the future" (Idea Group Inc. Hershey, PA, USA; 2004). He is Vice-Chair of the Working Group 8.3 on DSSs of the International Federation for Information Processing (IFIP).

Peter Braun is a post-doctoral research fellow in the Faculty of Information and Communication Technologies, Swinburne University of Technology, Australia and member of the Centre for Intelligent Agents and Multi-Agent Systems. He received a Diploma degree in computer science in 1998 and a PhD degree in computer science from Friedrich Schiller University Jena, Germany, in 2003. His research interests include distributed and mobile systems, in particular intelligent and mobile agent systems.

Jakub Brzostowski is currently a PhD Student at the Faculty of Information and Communication Technologies, Swinburne University of Technology (Melbourne, Australia). He is member of the Centre for Intelligent Agents and Multi-Agent Systems (CIAMAS). Before that, he has been a faculty member in the Centre of Applied Mathematics and Mathematical Modelling of the Mathematics and Physics Faculty at the Silesian University of Technology, Poland, where he was working from September 2002 till February 2004. He received a MSc degree in mathematics at SUT Gliwice in June 2002. His research interests involve automated negotiation, qualitative decision making, possibility theory and theory of fuzzy sets.

Frada Burstein is Associate Professor at Monash University. She gained her PhD in Decision Support Systems from the Soviet Academy of Sciences. At Monash University she has set up and leads a Knowledge Management Laboratory for studying modern technologies for supporting knowledge creation, storage, communication and application. She is a member of the Editorial Board of the Journal of Information and Knowledge Management and the Journal of Decision Systems, Associate Editor for the International Journal of Knowledge Management and Decision Support Systems journal. Frada was a Chief Investigator and a system architect for the Breast Cancer Knowledge Online (BCKO) project.

Nathalie Cassaigne is a lecturer of Decision Technologies at The University of Manchester, UK. She holds two post-graduate degrees in Computation and in Management Science from the University Toulouse 1 (France). She has over sixteen years experience in helping senior executives unlock corporate and personal benefits by supporting and optimizing strategic and tactical decisions. She has operated in a wide range of organizations in the private and public sectors including software, telecommunication, manufacturing, aerospace industries and government. Her major research interest is the application of decision technologies to tactical and strategic business decision-making from a systems engineering perspective. She is a reviewer and Associate Editor of many international journals including the IEEE Transactions on Systems, Man and Cybernetics.

Francisco Cervantes-Pérez is Director of the Center for Open University and Distance Learning at the National Autonomous University of Mexico (UNAM). He received his BS in Mechanical Electrical Engineering (1978) and his MEng (1982) from the National Autonomous University of Mexico, and his PhD in Computer and Information Sciences from the University of Massachusetts, at Amherst (1985). His research interests are: first, how the brain works during sensory-motor coordination, via mathematical modeling and computer simulations; and, second, the development of automata, based on Artificial Intelligence, Neuronal Networks and Schema Theory, to solve complex problems in robotics, pattern recognition and decision-making support systems.

Nabie Y. Conteh is a Systems Engineer with emphasis on Decision Support Systems, a researcher at the University of Maryland Baltimore County (UMBC) and an Assistant Professor in Software Engineering at the University of Maryland University College. Dr. Conteh holds a B.S in Information systems, an MBA in information systems management from Ferris State University in Michigan, a master’s degree and a PhD in Information Systems from the University of Maryland, Baltimore County. He has written nine papers, including three book chapters, three journal publications and in addition, presented three in top conference proceedings.

Bertha A. Correa-Pérez is an Adjunct Professor of MIS at the College of Business Administration at Florida International University (FIU). She joined FIU in August 2000 as a Project Manager and Research Assistant at the FIU/NASA Knowledge Management Laboratory. She has also served as a professor of IT and as a consultant for the private sector in Venezuela. Her research interests include knowledge management, artificial intelligence systems, decision support systems, and the impact of Information Technology on the business processes. Bertha

Correa graduated from FIU with a Master of Sciences in Management Information Systems and a BS in Computer Sciences. She also holds a degree as Computer Engineering Technician from Instituto Universitario de Nuevas Profesiones in Venezuela.

James F. Courtney is Professor of Management Information Systems at the University of Central Florida in Orlando. He formerly was Tenneco Professor of Business Administration in the Information and Operations Management Department at Texas A&M University. He received his PhD in Business Administration (with a major in Management Science) from the University of Texas at Austin in 1974. His papers have appeared in several journals, including Management Science, MIS Quarterly, Communications of the ACM, IEEE Transactions on Systems, Man and Cybernetics, Decision Sciences, Decision Support Systems, the Journal of Management Information Systems, Database, Interfaces, the Journal of Applied Systems Analysis, and the Journal of Experiential Learning and Simulation. His present research interests are knowledge-based decision support systems, ethical decision making, knowledge management, inquiring (learning) organizations and sustainable economic systems.

Omar El-Gayar is an Associate Professor of Information Systems at the College of Business and Information Systems, Dakota State University. His research interests include: decision support systems, multiple-criteria decision making, the application of information technology in aquaculture, agriculture, and natural-resource management, software engineering, and software process simulation modeling. He has been involved in the development of a number of multiple-criteria decision support systems including aquaculture and fisheries. He has an interdisciplinary educational background and training in information technology, computer science, economics, and operations research. He has numerous publications in the various information technology related fields.

Hamid R. Ekbia, PhD, is currently an Assistant Professor of computer science at the University of Redlands. His research focuses on how knowledge is developed, transformed, mobilized, and shared in and among individuals, organizations, and communities. He has written about these topics in Artificial Intelligence, Decision Support Systems, Knowledge Management, and Free/Open Source Software development.

Julie Fisher works in the School of Information Management and Systems at Monash University and has worked in the information systems field for the last 15 years. Her research interests have most recently focused on the area of human-computer interaction and in particular usability. Julie is also interested generally in the systems development process and how it can be improved to produce better systems from a user perspective. As a Chief Investigator Julie has been responsible for the design and usability study of the BCKOnline portal prototype.

Guisseppi A. Forgionne is Professor of Information Systems at the University of Maryland Baltimore County (UMBC). Professor Forgionne holds a BS in

Commerce and Finance, an MA in Econometrics, an MBA, and a PhD in Management Science and Econometrics. He has published 25 books and approximately 175 research articles and consulted for a variety of public and private organizations on decision support systems theory and applications. Dr. Forgionne has also served as department chair at UMBC, Mount Vernon College, and Cal Poly Pomona. He has received several national and international awards for his work.

Alexandre Gachet is a visiting research scholar in the Information Technology Department at the University of Hawaii at Manoa. He holds a diploma and a PhD in computer science from the University of Fribourg, Switzerland. His research interests include the design and implementation of distributed decision support systems using new distributed technologies, with a focus on highly decentralized, self-managed, and self-healing architectures.

Leonardo Garrido is professor and researcher of the Center for Intelligent Systems at the Monterrey Institute of Technology (ITESM). He holds a MS in Computer Systems and a PhD in Artificial Intelligence (with specialization in Autonomous Agents and Multiagent Systems). During his doctoral studies he was a visiting research scholar in the Robotics Institute at Carnegie Mellon University (CMU) for three years where he conducted research on Multiagent meeting scheduling and learning models about other agents in Multiagent environments. He has published in several international conferences specialized in Agents, Multiagent Systems and Artificial Intelligence. During the most recent years, he has served as a member of program and scientific committees of several conferences, such as the International Conference on Autonomous Agents and Multiagent Systems (AAMAS) and the Mexican International Conference on Artificial Intelligence (MICAI). He has also served as external reviewer for several top journals and conferences.

Ovsei Gelman is a Senior Researcher at the Center of Applied Sciences and Technology Development (CCADET) of the National Autonomous University of Mexico (UNAM). He holds a BS, MS. and PhD in Physics and Mathematics from the University of Tbilisi, Georgia. In the last 35 years, Dr. Gelman has contributed to the advance of the Systems Science discipline, through the publication of approximately 250 research papers in books, national and international journals and congress proceedings, the conduction of consulting activities for governmental and private organizations and his participation as advisor in the Graduate University Program of Engineering at UNAM.

Carina S. Gonzalez (cjgonza@ull.es) was born in Corrientes, Argentina (1971); she has the degrees BS in Computer Science (1995) and PhD in Computer Science (2001) with specialty in Artificial Intelligence (IA) and Intelligent Multimedia Interfaces. She continues working on these topics in several research projects and has more than thirty papers in conferences and journals. Some fellowships that she had were: PhD fellowship MUTIS (1997-2000), Carnegie Mellon University

(2002), PLANET European Network of excellence in Planning (2002). Since 2002 she has been Associated Professor of Computer Science, University of La Laguna.

Michael Goul is a Professor of Information Systems. His research has been published in a wide range of both academic and practitioner journals, and his recent research interests are in the areas of enterprise modeling, information systems strategy, data warehousing, decision support, knowledge management systems and services-oriented systems development. Dr. Goul has published in journals including Decision Support Systems, Journal of Management Information Systems, Electronic Commerce, Information and Management, Decision Sciences, Information Systems Frontiers and other journals. He has served as a journal editor, special issue coeditor, Association for Information Systems (AIS) Council Vice-President, AIS Conference Chair and Program Chair, Director of the highly ranked Information Management specialization for ASU's Master of Business Administration Program and Computer Information Systems Doctoral Program Coordinator. He is currently the Chair of the AIS Special Interest Group in Decision Support, Knowledge and Data Management Systems.

Jatinder N.D. Gupta is currently Eminent Scholar of Management of Technology, Professor of Management Information Systems, Industrial and Systems Engineering and Engineering Management, and former Chairperson of the Department of Accounting and Information Systems at the University of Alabama in Huntsville, Huntsville, Alabama. Most recently, he was Professor of Management, Information and Communication Sciences, and Industry and Technology at the Ball State University, Muncie, Indiana. He holds a PhD in Industrial Engineering (with specialization in Production Management and Information Systems) from Texas Tech University. Coauthor of a textbook in Operations Research, Dr. Gupta serves on the editorial boards of several national and international journals. Recipient of the Outstanding Faculty and Outstanding Researcher awards from Ball State University, he has published numerous papers in such journals as Journal of Management Information Systems, International Journal of Information Management, Operations Research, INFORMS Journal of Computing, Annals of Operations Research, and Mathematics of Operations Research. More recently, he served as a coeditor of several special issues including the Neural Networks in Business of Computers and Operations Research and books that included Decision Making Support Systems: Achievements and Challenges for the New Decade published by Idea Group Publishing. His current research interests include e-commerce, supply-chain management, information and decision technologies, scheduling, planning and control, organizational learning and effectiveness, systems education, knowledge management, and enterprise integration. Dr. Gupta has held elected and appointed positions in several academic and professional societies including the Association for Information Systems, Production and Operations Management Society (POMS), the Decision Sciences Institute (DSI), and the Information Resources Management Association (IRMA).

Pius Haettenschwiler is an assistant professor in the Department of Computer Science at the University of Fribourg, Switzerland. His research interests include knowledge-based systems (with a focus on the mathematical modeling of business logic), as well as concepts, tools and frameworks for DSS development. Dr. Haettenschwiler has extended experience in developing DSS for governmental purposes, such as food security and agricultural policy.

Nikos Karacapilidis holds an Associate Professor position at the Industrial Management and Information Systems Lab, Department of Mechanical Engineering and Aeronautics, University of Patras, Greece. His current research interests are on the areas of Intelligent Information Systems, Decision Making, Knowledge Management, and Semantic Web. Information about his professional activities and publication list can be found at http://www.mech.upatras.gr/~nikos/.

Gregory E. Kersten is a professor of decision and information systems at the John Molson School of Business, Concordia University. He received his M.Sc. in Econometrics and a PhD in Operations Research from the Warsaw School of Economics, Poland. In 1996 he set up the InterNeg Project; since 2002 he has been the Director of the E-negotiation, media and transaction research program. He authored and coauthored 3 books, 25 chapters in books, and over 100 refereed papers.

Jin Baek Kim is an assistant professor of Quality Systems Engineering at Concordia Institute of Information Systems Engineering, Concordia University. He is a member of the InterNeg research group and received a post-doctoral fellowship from the Social Sciences and Humanities Research Council of Canada (SSHRC). He received his BS in Electrical Engineering from Seoul National University, and MS and PhD degrees in Industrial Engineering and Operations Research from U. C. Berkeley. His research interests include decision support systems, supply-chain management, auctions, and e-negotiation systems.

Woo-Je Kim is a Professor of Industrial and Information Systems Engineering at Seoul National University of Technology, Seoul, Korea. He received his PhD in operations research at Seoul National University and has worked at the University of Michigan as a visiting scholar. He is interested in the application of optimization models in logistics and software engineering.

Ryszard Kowalczyk is Professor of Intelligent Systems and Director of the Centre for Intelligent Agents and Multi-Agents Systems in the Faculty of Information and Communication Technologies, Swinburne University of Technology, Australia. He has some 20 years of combined academic and industrial research experience in intelligent systems and their applications in a wide range of complex real-world problems. Ryszard has authored 4 patents and some 100 publications. His research interests involve autonomous negotiation, coordination and adaptation mechanisms in intelligent agent systems.

Soon-Jae Kwon is a lecturer of MIS at Sungkyunkwan University in Seoul, Korea. He received his PhD in MIS from Sungkyunkwan University, a Master of business administration and a BA in Accounting from the same university. His research focuses on decision analysis in electronic commerce issues. He is preparing working papers about the flow of Website with applications to B2B. His research works have appeared in Journal of Management Information Systems, Expert Systems with Applications

Karl Lang holds a PhD in Management Science from the University of Texas at Austin and is currently Associate Professor in Information Systems at Baruch College in New York City. Dr. Lang's research interests include decision technologies, management of digital businesses, knowledge-based products and services and issues related to the newly arising informational society. His findings have been published in such diverse journals as Communications of the ACM, Journal of Management Information Systems, Decision Support Systems, Computational Economics and Annals of Operations Research. He is also Associate Editor of the Decision Support Systems journal.

Kun-Chang Lee is a Full Professor of MIS and AIS at Sungkyunkwan University in Seoul, Korea. He received his PhD in MIS from Korea Advanced Institute of Science and Technology (KAIST), a Master of Sciences in MIS from KAIST, and a BA in business administration from Sungkyunkwan University, Seoul, Korea. His recent research interests lie in cognitive map analysis of electronic commerce issues. Recently, he is developing several working papers specializing in multi-agent cognitive maps, knowledge management, and ubiquitous computing with applications to B2B and B2C problems. His research findings have been published in Journal of Management Information Systems, Decision Support Systems, International Journal of Production Research, Expert Systems with Applications, Fuzzy Sets and Systems, Intelligent Systems in Accounting Finance and Management, Computers & OR, Computers & IE, Simulation, Expert Systems, Computers in Human Behavior, among others.

Carlos A. Legna holds a PhD in economics (Pierre Mendès France University, Grenoble, France) and was an expert of the United Nation technical cooperation for the development (Department of Economic and Social Affairs), working in Latin America and Africa. He has specialized in development theory and planning and public decision making, particularly in respect to underdeveloped countries and regions. At present, he is professor of Strategic Regional Planning, Economic Regional Policies, and Evaluation of Social Project at La Laguna University, Canary Islands, Spain. He has published recently the book "Gestión Pública Estratégica y Prospectiva", con aplicaciones al ámbito regional, among other books and articles.

Lionel Lorimier received a MS Degree in control theory from the University of Lille, France in 2002, an Engineering Degree in control science and advanced mechanics in 2002 and a PhD in control theory and computer science in 2005, both from the Ecole Centrale de Lille, France. His PhD dealt with the design of an innovative strategy to manage unknown failures online in complex engineered systems such as nuclear power plants and mobile robots. Concurrently, since 2001, he has worked on the application of control theory and artificial intelligence to decision theory with the University of Manchester, UK. His current research interests focus on i-DMSS at tactical and strategic levels, especially when many decision makers and experts are involved in the decision process.

Dr. Jinwei Lu is a Research Fellow at The University of Birmingham. He holds a BSc in Mechanics, a MEng in Mechanical Engineering and a PhD in Systems Engineering. Prior to joining The University of Birmingham, he worked at China's Southeast University, Hong Kong University of Science and Technology and The University of Leeds, UK. His research interests include engineering dynamics, control and decision support. He has published over twenty papers in conferences and journals such as Vehicle System Dynamics.

C. Gustav Lundberg (PhD, SUNY Buffalo) is Professor of Management Science and Information Systems Management at A.J. Palumbo School of Business Administration and John F. Donahue Graduate School of Business of Duquesne University in Pittsburgh, Pennsylvania, USA, and Docent at the Swedish School of Economics and Business Administration in Helsinki, Finland. Some of his recent publications have appeared in European Journal of Operational Research, Journal of Economic Psychology, and Scandinavian Journal of Psychology.

Keith Madelin is a Professor in Civil Engineering in the School of Engineering at The University of Birmingham where he has led the railway research group for some eight years. He was Director of Railway Research UK and chairman of the Council for the European Railway Research Network of Excellence in (EURNEX). His main research interests lie in the fields of railway track, asset management and systems performance.

Pooja Malhotra entered the field of Information Systems after having worked as an IT Professional for five years. She has degrees in Graduate Diploma in Information Management and Systems from Monash University, Bachelor of Commerce from Mumbai University, Diploma in Information Technology from National Institute of Information Technology, Mumbai. Her interests lie in information and knowledge management and also human factors in computer systems design. She is currently working as a business analyst in the finance and banking industry and pursuing her Masters in Information Systems. BCKOnline portal prototype design comprises a part of her research project.

Rosetta Manaszewicz is currently a PhD candidate at the School of Inforamtion Management and Systems Monash University. She is also a breast cancer consumer advocate and has been part of numerous national and state working parties and committees that include: the Standards Development Management Committee (Victorian Cancer Co-ordination Unit); Committee member Victorian Clinical Oncology Group (Cancer Council of Victoria); and editor of the Breast

Cancer Action Group newsletter. Rosetta has also been involved with the development and evaluation of the BCKOnline portal prototype.

Andrew McCosh is the Alvah Chapman Eminent Scholar in Management and Ethics at Florida International University. His 1999 book on Financial Ethics predicted that the pressures on the executive ranks to produce ever-greater performance might lead to serious ethical difficulties and perhaps to excessively creative accounting. He is also Professor of Finance Emeritus at Edinburgh University, UK, and has served as a director of four mutual fund families. He was Dean of the Business School at Manchester University (England) and at Edinburgh University (Scotland) before joining FIU. He holds five degrees in various subjects ranging from mathematics to theology.

Sue McKemmish is Chair of Archival Systems at Monash University. She is engaged in major research projects and standards initiatives relating to the clever use of metadata, archival description, resource discovery, and the nexus between memories, communities and technologies. She leads the Australian Team in the International Project relating to the Preservation of Authentic Records in Electronic Systems, and is co-Director of the Metadata and Archival Description Registry and Analysis System project. She has a PhD in archival systems and is a Laureate of the Australian Society of Archivists. As a Chief Investigator for the BCKO project, Sue focused on metadata related issues.

Manuel Mora is an Associate Professor of Information System at the Autonomous University of Aguascalientes, México (UAA). He holds a doctoral degree in Engineering from the National Autonomous University of Mexico (UNAM), and a MSc and a BS in Computer Systems from Monterrey Tech (ITESM). His main research interests are the Modeling of Complex Systems using the Systems Engineering and Systems Approach and the Theoretical Foundations for Decision-Making Support Systems and Information Systems. He has been coeditor of a book and three special issues on DMSS for international journals and has published research works in several international settings like IRMA, AMCIS, KES and IFIP Conferences. He is a member of AIS, SIGDSS, ACM and INCOSE.

Katta G. Murty is a Professor of Industrial and Operations Engineering at the University of Michigan, Ann Arbor, Michigan, USA. He teaches courses in Operations Research, in particular those dealing with optimization models for decision making and their applications. He has made important contributions for the development of efficient algorithms to several optimization models, and in applying these algorithms intelligently to real-world problems. He is a Fellow of INFORMS.

Robbie T. Nakatsu is an Assistant Professor at the College of Business Administration, Loyola Marymount University, Los Angeles, California. He received his PhD (2001) in Management Information Systems from the University of British Columbia. His research interests include intelligent systems, human-computer interaction, and computer simulation modeling.

Jason Nichols is a doctoral student in the Information Systems Department of the W.P. Carey School of Business at Arizona State University. He has an MBA and a Masters of Science in Information Management. Jason has experience in software development, project management, professional education and training, and information systems consulting. His research interests include service-oriented architecture, information systems strategy, knowledge management, and decision support. He has been a publisher in the proceedings of the Hawaii International Conference on System Sciences and has a forthcoming paper in IEEE Transactions on Systems, Man, and Cybernetics.

David Paradice is Professor and Chairman of the MIS Department at Florida State University. He received his PhD in Business Administration (Management Information Systems) from Texas Tech University. He has worked as a programmer analyst and consultant. Dr. Paradice has published numerous articles focusing on the use of computer-based systems in support of managerial problem formulation and on the influence of computer-based systems on ethical decision-making processes. His publications appear in Journal of MIS, IEEE Transactions on Systems, Man & Cybernetics, Decision Sciences, Communications of the ACM, Decision Support Systems, Annals of Operations Research, Journal of Business Ethics, and other journals.

Gloria Phillips-Wren is assistant professor in the Information Systems and Operations Management department at Loyola College in Maryland. She joined the faculty after a career in research and development for the US government where she was awarded a Superior Civilian Service Award. She received MS and PhD degrees in management information systems from the University of Maryland Baltimore County, and is coeditor of a recently-published book on *Intelligent Decision Support Systems in Agent-Mediated Environments.* Her current research interests are development and evaluation of decision support systems, intelligent agents, datamining, and global information systems. She is a member of AIS, DSI, INFORMS and IRMA.

Giuliano Pistolesi has worked in computational simulation of cognitive systems by AI and ALife techniques for about 11 years. He holds a MSc in Cognitive Psychology and a PhD in Medical Informatics from the University of Rome "La Sapienza". At ThinkinGolem he is primarily involved in several R&D projects regarding intelligent user interfaces, mobile software and natural language processing systems. His interests range from Evolutionary Computation to Cognitive Science and Technology, especially Intelligent Agents and Human-Machine Interaction. He has published in several scientific journals, including Cybernetica and IEEE: Transactions on Reliability. His background is both in Computer Science and Cognitive Psychology.

Jean-Charles Pomerol is Professor of Computer Science and Vice-Rector in charge of Research at University Pierre and Marie Curie (UPMC) in Paris (France). For six years he headed UPMC's Artificial Intelligence Laboratory and then for five years the Office of Technology Transfer. He was also with the French National Center for Scientific Research (CNRS) as a project manager for information science and technology from 1995 to 2000. Professor Pomerol has published a number of books on Expert Systems, Decision Support Systems and multicriterion decision making and numerous papers in international journals. In 1992, he launched the Journal of Decision Systems, which quickly became a successful journal in the DSS area. He was the Editor-in-Chief of JDS until he stepped down in 2002. He still is the Editor-in-Chief of the “Revue Française d’Intelligence Artificielle”.

Clive Roberts is Lecturer in Railway Systems at The University of Birmingham and is a member of Railway Research UK and the European Railway Research Network of Excellence in (EURNEX). His main research interests lie in the fields of railway operations and systems performance, in particular: railway condition monitoring, systems engineering, data management and system modeling and optimization. He lectures in Engineering Mathematics, Electrical Engineering and Systems Engineering at both The University of Birmingham and The University of Sheffield.

Ram D. Sriram currently leads the Design and Process Group in the Manufacturing Systems Integration Division at the National Institute of Standards and Technology, where he conducts research on standards for interoperability of computer-aided design systems. He attained a BS from IIT, Madras, India, and an MS and a PhD from Carnegie Mellon University, Pittsburgh, PA. From 1986 to 1994 he was on the Engineering Faculty at the Massachusetts Institute of Technology (MIT) and was instrumental in setting up the Intelligent Engineering Systems Laboratory. At MIT Ram initiated the MITDICE project, which was one of the pioneering projects in collaborative engineering. His work on the MIT-DICE project was compiled into a book. Dr. Sriram has coauthored or authored more than 180 papers and reports and 13 books on computer-aided engineering. He was a Founding Co-editor of the International Journal for AI in Engineering. In 1989 he was awarded a Presidential Young Investigators Award from the US National Science Foundation

Alan Stirling manages the Railway Research Group at The University of Birmingham and the headquarters of Railway Research UK, the UK universities’ center for railway systems research. His main research interests lie in the fields of railway track, asset management and whole systems performance. He is a chartered civil engineer with a background in highway engineering and has spent the past eight years undertaking various railway research projects at the University of Birmingham.

Stefan Strecker is a postdoctoral fellow at the Institute for Computer Science and Business Information Systems (ICB), University Duisburg-Essen, Germany. At the time of writing, he was a postdoctoral fellow with the SSHRC project "Electronic negotiations, media, and transactions in socio-economic interactions" at the John Molson School of Business, Concordia University, Montreal, Quebec, Canada. He received his doctorate in Business Administration from the Department of Economics and Business Engineering at the Universitaet Karlsruhe (TH) in Germany. His research interests include electronic markets and negotiations.

Kanchana Tandekar holds a Bachelor of Engineering Degree in Computer Science from Rajiv Gandhi Technical University, Bhopal, India. She also holds a Master of Science Degree in Information Systems from Dakota State University, Madison, South Dakota. She's currently working as a Programmer/Systems Analyst at Factor 360, a Web design and development company in Pierre, SD. Her research interest includes Decision Support Systems, Artificial Intelligence, Model Management systems, Data Mining and Warehousing.

Rustam Vahidov is an associate professor of Management Information Systems at the Department of Decision Sciences and MIS at John Molson School of Business, Concordia University, Montreal, Quebec (Canada). He earned his PhD and MBA degrees from Georgia State University (Atlanta, Georgia). Dr. Vahidov's research interests include decision and negotiation support, design research, agent-based systems, genetic algorithms, fuzzy logic, neural networks, and electronic commerce. His works have been published in Journal of Management Information Systems, Journal of Decision Support Systems, Information & Management, International Journal of Intelligent Systems, Fuzzy Sets and Systems, and other journals, edited books, and conference proceedings.

Sérgio Ivan Viademonte da Rosa. He holds a PhD degree in Computing, Monash University, Australia (2004); MSc degree in Information Systems, Federal University of RS, Brazil (1995); and a Bachelor degree in Computer Science from Pontifical Catholic University of RS, Brazil (1989.) He has been assigned to Information Technology positions in Australia, as a research fellow at the Univeristy of Constance, Germany. He has been lecturing computer science subjects at Monash University (Australia), Pontificial Catholic University RS (Brazil) and Feevale Graduate School (Brazil). He was one of the directors and founders of a start up software company in Brazil. Areas of interest include intelligent systems and knowledge discovery.

Gerald R. Wagner is a Distinguished Research Fellow, Peter Kiewit Institute, University of Nebraska, Omaha. He is also the faculty director of One Innovation Place and a Gallup Senior Scientist. Prior to joining the faculty at Nebraska he started several software companies and was coordinator of the Operations Research Group, College of Engineering, University of Texas at Austin. His most notable software development undertaking was a product called the Interactive Financial Planning System© (IFPS) that was used by several academic institutions in teaching and research. His current development of the Planners Lab© will follow the same academic model.

Yun Wan is an Assistant Professor of Computer Science and Information System in the University of Houston, Victoria. His research interests include electronic commerce, information system analysis and design, and decision support systems. He has published several peer-reviewed papers about comparison-shopping agent in the past two years. He had presented in several international conferences, *e. g.* International Conference on Electronic Commerce. He is a Fellow of 2003 Doctoral Consortium of International Conference on Information Systems. He obtained his DBA in Management Information System from Liautaud Graduate School of Business, University of Illinois at Chicago.

Xuan F. Zha is currently with the Design and Process Group, Manufacturing System Integration Division, National Institute of Standards and Technology (NIST), USA. He is also a Guest Professor of Shanghai JiaoTong University, China. He received his PhD from Nanyang Technological University, Singapore. He is a Senior Member of IEEE and SME. His current research interests are in virtual and collaborative product design and development, product lifecycle management, knowledge management in design, artificial intelligence and soft computing in design and manufacturing, supply-chain management, micro/nanoelectro-mechanical systems (MEMS/NEMS) design and simulation, embedded systems, robotics, and so forth. Dr. Zha has authored or coauthored over 100 papers, book chapters, and reports in these areas.

Index