Studies in Computational Intelligence 97

Gloria Phillips-Wren
Nikhil Ichalkaranje · Lakhmi Jain
(Eds.)

Intelligent Decision Making: An AI-Based Approach

Springer

Gloria Phillips-Wren, Nikhil Ichalkaranje and Lakhmi C. Jain (Eds.)

Intelligent Decision Making: An AI-Based Approach

Studies in Computational Intelligence, Volume 97

Editor-in-chief
Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

Further volumes of this series can be found on our homepage: springer.com

Vol. 74. Robert Schaefer
Foundation of Global Genetic Optimization, 2007
ISBN 978-3-540-73191-7

Vol. 75. Crina Grosan, Ajith Abraham and Hisao Ishibuchi (Eds.)
Hybrid Evolutionary Algorithms, 2007
ISBN 978-3-540-73296-9

Vol. 76. Subhas Chandra Mukhopadhyay and Gourab Sen Gupta (Eds.)
Autonomous Robots and Agents, 2007
ISBN 978-3-540-73423-9

Vol. 77. Barbara Hammer and Pascal Hitzler (Eds.)
Perspectives of Neural-Symbolic Integration, 2007
ISBN 978-3-540-73953-1

Vol. 78. Costin Badica and Marcin Paprzycki (Eds.)
Intelligent and Distributed Computing, 2008
ISBN 978-3-540-74929-5

Vol. 79. Xing Cai and T.-C. Jim Yeh (Eds.)
Quantitative Information Fusion for Hydrological Sciences, 2008
ISBN 978-3-540-75383-4

Vol. 80. Joachim Diederich
Rule Extraction from Support Vector Machines, 2008
ISBN 978-3-540-75389-6

Vol. 81. K. Sridharan
Robotic Exploration and Landmark Determination, 2008
ISBN 978-3-540-75393-3

Vol. 82. Ajith Abraham, Crina Grosan and Witold Pedrycz (Eds.)
Engineering Evolutionary Intelligent Systems, 2008
ISBN 978-3-540-75395-7

Vol. 83. Bhanu Prasad and S.R.M. Prasanna (Eds.)
Speech, Audio, Image and Biomedical Signal Processing using Neural Networks, 2008
ISBN 978-3-540-75397-1

Vol. 84. Marek R. Ogiela and Ryszard Tadeusiewicz
Modern Computational Intelligence Methods for the Interpretation of Medical Images, 2008
ISBN 978-3-540-75399-5

Vol. 85. Arpad Kelemen, Ajith Abraham and Yulan Liang (Eds.)
Computational Intelligence in Medical Informatics, 2008
ISBN 978-3-540-75766-5

Vol. 86. Zbigniew Les and Mogdalena Les
Shape Understanding Systems, 2008
ISBN 978-3-540-75768-9

Vol. 87. Yuri Avramenko and Andrzej Kraslawski
Case Based Design, 2008
ISBN 978-3-540-75705-4

Vol. 88. Tina Yu, David Davis, Cem Baydar and Rajkumar Roy (Eds.)
Evolutionary Computation in Practice, 2008
ISBN 978-3-540-75770-2

Vol. 89. Ito Takayuki, Hattori Hiromitsu, Zhang Minjie and Matsuo Tokuro (Eds.)
Rational, Robust, Secure, 2008
ISBN 978-3-540-76281-2

Vol. 90. Simone Marinai and Hiromichi Fujisawa (Eds.)
Machine Learning in Document Analysis and Recognition, 2008
ISBN 978-3-540-76279-9

Vol. 91. Horst Bunke, Kandel Abraham and Last Mark (Eds.)
Applied Pattern Recognition, 2008
ISBN 978-3-540-76830-2

Vol. 92. Ang Yang, Yin Shan and Lam Thu Bui (Eds.)
Success in Evolutionary Computation, 2008
ISBN 978-3-540-76285-0

Vol. 93. Manolis Wallace, Marios Angelides and Phivos Mylonas (Eds.)
Advances in Semantic Media Adaptation and Personalization, 2008
ISBN 978-3-540-76359-8

Vol. 94. Arpad Kelemen, Ajith Abraham and Yuehui Chen (Eds.)
Computational Intelligence in Bioinformatics, 2008
ISBN 978-3-540-76802-9

Vol. 95. Radu Dogaru
Systematic Design for Emergence in Cellular Nonlinear Networks, 2008
ISBN 978-3-540-76800-5

Vol. 96. Aboul Ella Hassanien, Ajith Abraham and Janusz Kacprzyky (Eds.)
Computational Intelligence in Multimedia Processing: Recent Advances, 2008
ISBN 978-3-540-76826-5

Vol. 97. Gloria Phillips-Wren, Nikhil Ichalkaranje and Lakhmi C. Jain (Eds.)
Intelligent Decision Making: An AI-Based Approach, 2008
ISBN 978-3-540-76829-9

Gloria Phillips-Wren
Nikhil Ichalkaranje
Lakhmi C. Jain
(Eds.)

Intelligent Decision Making: An AI-Based Approach

With 107 Figures and 44 Tables

Springer

Prof. Dr. Gloria Phillips-Wren
Information Systems and Operations Management
The Sellinger School of Business and Management
Loyola College in Maryland
4501 N. Charles Street
Baltimore, MD 21210
USA

Dr. Nikhil Ichalkaranje
School of Electrical
and Information Engineering
University of South Australia
Adelaide
South Australia SA 5095
Australia

Prof. Dr. Lakhmi C. Jain
School of Electrical and Information Engineering
University of South Australia
Adelaide
South Australia SA 5095
Australia

ISBN 978-3-540-76828-9 e-ISBN 978-3-540-76829-6

Studies in Computational Intelligence ISSN 1860-949X

Library of Congress Control Number: 2007939887

Cover Design: Deblik, Berlin, Germany

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

Dedicated to our mothers who were our first and best teachers.

Preface

The fusion of artificial intelligence (AI) with decision support systems (DSSs) is opening exciting new areas of research and application. The resulting systems are smarter, more efficient, adaptable, and better able to aid human decision making. While AI aims to mimic human behaviour in limited ways, DSSs attempt to help humans make the best choice among a set of possible choices given explicit or implied criteria. Long a topic of science fiction, AI today is demonstrating that it can be integrated effectively into real systems and that it offers the only way possible to capture aspects of human intelligence such as learning. The combination of AI and DSSs provides formidable new computational assistants to humans that extend their capabilities in routine and complex stressful environments. Due to the increasing maturity of this interdisciplinary field as evidenced by the recent growth in the number of research publications and contributors entering the field, a book that explores the current state and future outlook of intelligent DSSs seems appropriate.

The book is organized around three themes. The first two chapters provide a solid foundation by exploring studies and theories of human decision making. They trace some one hundred years of research including recent work by the well-known authors and provide a vision of the use of computerized decision aids. The second section deals with paradigms and methods associated with AI in DSS. The final section provides sample applications among the many that are appearing today and gives our perspective on future research directions needed to advance the field.

This book would not have been possible without the efforts of many people. We thank the contributors for their inspiring research and the reviewers for their efforts to create a high-quality book. The publisher's support, patience and assistance are gratefully acknowledged. In particular, Srilatha Achuthan's unwavering efforts as project manager provided help when we needed it most.

We thank the research community for the advances that have made this book possible and our families for their continued support.

USA *Gloria Phillips-Wren*
Australia *Nikhil Ichalkaranje*
Australia *Lakhmi C. Jain*

Foreword

Intelligent decision systems (IDS) are a relatively new paradigm in the decision support systems (DSS) area. Consistent with the modern view on work activity as mostly 'knowledge work' (Davenport, 2005) and recognising the critical role of knowledge for effective decision-making, intelligent decision support aims to provide the decision maker with quality assistance in gaining better knowledge and understanding of the decision situation. IDS are the means to achieve such assistance.

This need for knowledge management and processing within decision support systems has resulted in a special class of systems that incorporates qualitative knowledge and reasoning, extending the functionality beyond those traditionally covered by DSS applications. These systems, variously termed Intelligent Decision Support Systems, Intelligent Decision Systems, Knowledge-Based Decision Support Systems, Active DSS and Joint Cognitive Systems, include qualitative knowledge to extend the typically quantitative data of earlier approaches to decision support (Burstein and Holsapple, 2008; Gupta et al. 2006).

The label *intelligent* in IDS is derived from the attempts made in artificial intelligence (AI) to develop systems that computationally emulate some human cognitive capabilities such as reasoning, learning and memory. The need to incorporate domain knowledge and intelligent capabilities in decision support systems has been identified in various forms and models by many researchers, starting from Simon (1977), followed by Sprague (1993), and exemplified by Turban, Aronson and Liang (2005) and Holsapple and Whinston (1996) in their comprehensive analyses of tools and techniques for incorporating intelligence into DSS. Arnott and Pervan (2005), in their review of the DSS field, traced and described Intelligent Decision Support as a separate branch, which originated from research in AI and Expert Systems to complement the needs of modern Personalised Decision Support.

The main role of IDS in an organisation is as an enabler for knowledge processing with communication capabilities to support knowledge sharing and exchange and to facilitate organisational learning (Carlsson and Kalling, 2006;

Burstein and Linger, 2003). IDS aim to assist the decision maker in overcoming cognitive limitations to achieving the best decision outcomes. At the same time the system could identify some useful knowledge for future improvements in the decision-making process, thus facilitating continuous learning processes by an organisation. Conventional DSS was not intended to support such functionality, hence giving rise to IDS in a knowledge management context. Despite the significant potential of IDS and remarkable advances in AI technologies, the promise of IDS has not yet been realized.

IDS are not widespread as such. One reason is that comprehensive research is still required on AI technologies to be used in IDS. Some technologies such as intelligent agents have advanced to the point that they are implemented in numerous practical applications, while other AI concepts such as neural networks are not yet as mature. In most cases, specialized IDS applications are reported in the literature, although generalized applications have not been developed. Research is needed on architectures and frameworks that could support production-level IDS both at the AI and at systems levels. Although IDS do not in general exist as stand-alone systems, any large-scale management information system would include some intelligent components. Modern approaches to assisting organizations such as customer relationship management (CRM), knowledge management systems (KMS), and business intelligence (BI) systems are heavily influenced by intelligent techniques and include a wide range of intelligent systems functionality. Many such systems require access to expert or problem-domain knowledge. Availability of sophisticated generic technological infrastructure makes it easier to specialise such systems to suit specific application domains.

A number of books have been published in the area of IDS and related areas of Intelligent Decision Support Systems, and one needs to ask what another book can add to the community. Publication patterns over the last 10 years (shown in Fig. 1 based on data from Google Scholar) appear to show continued interest in IDS. This is a much needed book to update the interested reader in an exciting research field with many opportunities for advances in both theoretical and applied areas.

The current volume is an effort to bridge the range of exploration in this field from fundamental understanding of human decision making at an abstract conceptual level, to methods of computational intelligence, and to applications of intelligent decision support techniques in specific contexts. The book presents fascinating background information on human decision making and makes a contribution to the IDS area by presenting the current state of knowledge and identifying key research gaps. I would like to congratulate the editors of this book and look forward to it being remembered as a pivotal beginning for collective focus and mutual inspiration.

Victoria, Australia *Frada V. Burstein*

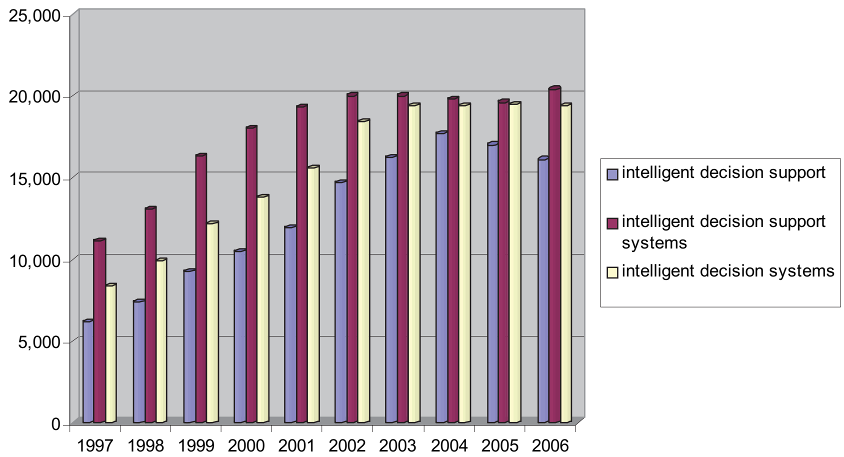


Fig. 1. Comparative data on publications in Intelligent Decision Support, Intelligent Decision Support Systems and Intelligent Decision Systems (based on the data from Google Scholar)

References

Arnott, D. and Pervan, G. (2005). A Critical Analysis of Decision Support Systems Research. *Journal of Information Technology*, 20, 67–87.

Burstein, F. and Holsapple, C.W. (eds.) (2008). *Handbook on Decision Support Systems*. Berlin/Heidelberg: Springer.

Burstein, F. and Linger, H. (2003). Supporting Post-Fordist Work Practices: A Knowledge Management Framework for Supporting Knowledge Work. *Information Technology and People*, 16, 3, 289–305.

Carlsson, S.A. and Kalling, T. (2006). Decision Support through Knowledge Management: What Works and What Breaks. In Adam, F., Brézillon, P., Carlsson, S. and Humphreys, P. (eds.), *Creativity and Innovation in Decision Making and Decision Support*, London, UK: Decision Support Press, 693–710.

Davenport, T.H. (2005). *Thinking for a Living: How to Get Better Performance and Results from Knowledge Workers*. Boston, MA: Harvard Business School Press.

Gupta J., Forgionne G., and Mora M. (eds.) (2006). *Intelligent Decision-Making Support Systems (i-DMSS): Foundations, Applications and Challenges*. Engineering Decision Series of Springer.

Simon, H. (1977). *The New Science of Management Decisions*. New Jersey, NJ: Prentice-Hall.

Sprague, R.H. (1993). A Framework for the Development of Decision Support Systems. In Sprague, R.H. and Watson, H.J. (eds.), *Decision Support Systems Putting Theory into Practice*, New Jersey, NJ: Prentice-Hall International, 3–26.

Turban, E., Aronson, J. and Liang, T. (2005). *Decision Support Systems and Intelligence Systems*. 7th Ed., New Jersey: Pearson Prentice-Hall.

Holsapple C. and Whinston, A. (1996). *Decision Support Systems: A Knowledge Based Approach*. St. Paul, MN: West Publishing.

Contents

Part I

Background: Human Decision Making

1

Understanding Human Decision Making – A Fundamental Step Towards Effective Intelligent Decision Support

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

[1] Laboratoire d'Informatique de Paris 6 (LIP6), Université Pierre et Marie Curie, 4 Place Jussieu, 75252 Paris 5, France, Jean-Charles.Pomerol@lip6.fr
[2] Business Information Systems, University College Cork, Cork, Ireland, fadam@afis.ucc.ie

Summary. As researchers try to accumulate knowledge in artificial intelligence (AI), towards developing better models and artefacts to embody complex decision making processes based on the characteristics of human decision making, we are reminded that at the beginning of this whole endeavour our intellectual ancestors – Newell and Simon (1972) for instance, had warned that a comprehensive understanding of human decision making would be required if AI was to yield substantial benefits. In wondering whether this has been achieved, we trace back the accumulated knowledge in the area of human decision making from the work of Savage through to that of Simon and we critically assess whether we have reached the required critical mass in our understanding of human decisions. Such knowledge development is a requisite benchmark to measure the progress of research in artificial intelligence, as illustrated by the other chapters in this book.

1.1 Introduction: Neurobiology of Human Reasoning and Decision Making

Although decision making is an activity that is almost as exclusively human as language itself[1], its neurobiological components have not been studied until the end of the twentieth century, which is comparatively much later than the investigation of the biology of language (Damasio 1994; Damasio et al. 1996).

Research in this critical area has generated two fundamental results. First of all, it has revealed the existence of a centralised area in the ventromedial prefrontal lobe of the brain where reasoned decision making takes place (Damasio 1994; Fuster 1996; Berthoz 2003). Any destruction or lesion in this area leads to highly irrational behaviour in previously "normal" subjects, as

[1] "Decision making is, in fact, as defining a human trait as language" (Damasio et al. 1996)

J.-C. Pomerol and F. Adam: *Understanding Human Decision Making – A Fundamental Step Towards Effective Intelligent Decision Support*, Studies in Computational Intelligence (SCI) **97**, 3–40 (2008)
www.springerlink.com

illustrated by the Phineas Gage case in Damasio (1994). One of the striking symptoms characterising subjects who have been injured in this area of the brain is their indifference to risk or at least, their inability to "properly" assess risk (Damasio 1994; Adolphs et al. 1996)[2]. An alternative hypothesis which has been put forward by Pomerol (1997b), but has not (yet) received empirical validation, is that these subjects may have lost their ability to arbitrate between short term and long term benefits or tradeoffs, thereby pursuing immediate satisfaction of their needs rather than future gains. This, of course, would tally up with the symptoms described in Damasio's and Adolphs et al.'s research, where subjects seem to be unable to properly take obvious risk factors into account. Indeed, this inability to anticipate risks has already been observed in other cases of frontotemporal mental deficiency (Schoenbaum et al. 1998; Berthoz 2003, p. 99).

Secondly, this research has shown the crucial role which emotions play in decision making. Damasio for instance, has gone as far as predicting that the role of reasoning in decision making would be found by future researchers to be less than is now thought. This is further discussed in Sect. 1.5 of this chapter, which is concerned with cognitive and decisional biases, in particular the *frame effect*. The reduced role of reasoning in human decision making is not necessarily a cause of concern for AI researchers, however, as although it is beyond debate that the emotional side of human nature has a strong effect on decision making activities, it does not mean that this aspect of human decision making is beyond modelling, as Simon (1995) has illustrated. Different models can be proposed to describe the effect of human emotion on decision making at a cognitive level, in the shape of short circuits or positive reinforcement. For instance, intuition or, intuitive decision making has been defined in previous research as an instantaneous, quasi automatic decision triggered by an affective, visual or sensorial stimulus. Klein (1993) went further when his studies of firemen and emergency response personnel led him to the concept of *recognition-primed decision*, where decisions are based on the recognition of previously known patterns and a solution is designed to match this pattern. Klein's work is crucial because it properly emphasises the importance of the *matching* aspect of decision making (see Berthoz 1996[3]).

These observations justify our belief that there are two key poles in decision making: *reasoning* and *recognition*, which are inextricably linked in the case

[2] "Subjects with VM (ventromedial) frontal lesions [...] invariably lose money on the task as a result of continuously choosing cards from the risky decks, even after they have had substantial experience with the decks, and have lost money on them. Interestingly, the VM frontal patients are quite aware that they are losing money, and some even figure out the fact that the decks from which they are choosing are likely to be more risky. None of this knowledge, however, appears to influence their abnormal behavior, and they continue to choose from risky decks despite continued losses" (Adolphs et al. 1996, p. 162)

[3] "The brain is a matching machine and a simulator of action, not a "representational" machine" (Berthoz 1996, p. 89)

of human decision making. However, one may wonder how specifically human such behaviour really is? It could also be hypothesized that this characteristic of decision making grew throughout natural evolution with the development of the frontal lobe, the most recent portion of the brain. The simple observation of Nature around us provides countless examples of decisions based on the recognition of stimuli with varying states of complexity, from the worm crawling away from a drop of acid to the sheep running away from the shadow of a plane mistakenly identified as a bird of prey. In the first instance, we can identify the increasing complexity of the pattern recognised (Berthoz 1996)[4], then, we move to the learning capacity identified in birds and mammals by Pavlov. Thus, to return to our initial questioning: is the behaviour of the dog fetching its lead when its master puts on his coat evidence of the premise of a reasoning capacity (on a lower level than those displayed by human agents, but reasoning nonetheless)?

It should further be noted that reasoning can only occur on a significant scale in the presence of memory. It is undeniable, as observed by Newell and Simon (1972), that intelligent information processing systems are all built around an apparatus that can capture and interpret stimuli, a number of specific memories and an apparatus for symbolic reasoning; indeed, this is a perfect description of the human brain. Thus, memory, reasoning and decision have evolved in tandem throughout human evolution. Of course, language can be added to this list insofar as it is very similar to decision making: both activities require the chaining of sounds, words and inflexions for language and of images, memories, facts and actions for decision making (Calvin 1991, 1994). The fact that case-based reasoning has been described as a language dedicated to decision making reinforces this point.

In this chapter, we review the two key aspects of decision making: reasoning and recognition. We review the classical models of previous researchers and evoke the arguments of their proponents and opponents. Finally, we examine recognition based decision making, reasoning based decision making and consider the cognitive biases that affect decision making, which takes us back to our discussion of the brain.

1.2 Procedural Rationality and Bounded Rationality

1.2.1 The Savage Model and Expected Utility

Even though Savage's (1954) model has been very well described in previous research, it is useful to go back to its key elements and to pragmatically examine its true meaning for a theory of action. Savage's (1954) model is primarily

[4] "But we have also proposed the idea that, (...) higher central loops that have increasingly gained complexity during evolution operate on another mode that we have called a projective process. In this mode, signals are processed in internal loops having no direct link with sensors" (Berthoz, 1996, p. 84)

important because it provides a formal and coherent framework to think about decision making. Savage rightly insists upon the crucial difference between the elements which the decision maker cannot control (Events, referred to as the set E) and the elements which he can control (Actions, referred to as the set A). Using a simple example, we can illustrate this difference with the story of the man going for a walk and considering whether to take his umbrella. Two actions are possible: (TU) and (NTU). For the sake of argument, we can also assume that only two events can occur during the walk: rain (R) or no rain (NR). We can then use the matrix in Table 1.1 to describe a function of A x E in the set of consequences C. Here the set of results is $\{-2,\ 1,\ 0,\ 2\}$.

Savage says that if the decision maker follows a coherent decision making path towards making a choice, then there is a set of probabilities and a utility function U such that the decision maker can seek to maximise his expected utility for the said probabilities. Savage's theorem is often used in reverse – i.e. to suggest which action maximises the expected utility of the decision maker given a set of known probabilities for possible events.

Savage's model formalises a number of key aspects of decision making even before one considers the theorem itself. Firstly it copper fastens the separation between events and outcomes. It is a fundamental point because most novice researchers of decision making "trip" at this first hurdle and confuse the skill of a decision maker with the lucky occurrence of a positive outcome. Indeed, human nature may push us to claim as evidence of good reasoning the fact that we took no umbrella and it did not rain, even though clouds are everywhere to be seen. Savage's theory, however, makes no mistakes: because you ignored the greater probability of rain and you were simply lucky not to get soaked! The theory appears kinder when considered in reverse: it admits the possibility that a sound decision should turn out to be a disastrous one. This aspect of the theory is closer to typical human understanding as many people are not slow to invoke bad luck in such cases. Thus, Savage's separation between events, actions and outcomes is probably, as humorously stated by Howard (1988), his most important contribution to decision making theory.

On second examination, however, one must wonder whether it is a realistic viewpoint to separate the universe in terms of actions and events. Indeed, some

Table 1.1. Example of a decisional matrix

	R	NR
TU	1	0
NTU	–2	2

actions modify future events. If a manager sets a price change for a product (Action), then the reactions of competitors (events) are clearly the result of the manager's action. Generally speaking, the separation of the decision maker and the environment (including the other actors around him or her) is nothing but a simplification of reality (see Boland 1979 for a well argued criticism). Savage's theory illustrates that such a separation, however simplistic, is a required hypothesis for whoever wants to propose a theory of decision making and of rationality.

Unfortunately, there are many cases when separating actions and events is not fruitful. Gilboa and Schmeidler (1995) provide two such examples. Their first example is that of a recruiter seeking to hire a sales representative. The actions are represented by the potential candidates that can be hired. Events, on the other hand, do not lend themselves to such modelling: they are represented by the qualities of the candidates, their honesty, their performance, etc. To properly describe such events, one would have to be cognisant of all the present and future capacities of each candidate. Thus, events are characterised by significant uncertainty which managers must reduce by collecting information and interviewing the candidates. This scenario is better analysed in terms of multi-criteria decision making as described in Pomerol and Barba-Romero (1993) for instance.

Gilboa and Schmeidler's second example is that of strategic decision making. In this case, the horizon of the decision maker is so long that events must be seen as long chains of consecutive events. The multiplicity of sub-events leads to a combinatory explosion of the number of events. The famous case study of the Bay of Pigs invasion provides an illustration of the difficulty in arbitrating the short term and long term objectives of such decision making. In such cases, it is simply impossible to consider all conceivable resulting events and the search is limited to a few scenarios some more likely than others. In the case of the Bay of Pigs, it is well understood that the scenario that actually unfolded was never contemplated by the Kennedy administration, or else, they would never have gone ahead! Savage's work is quite applicable to such situations, with the proviso that the complexity and interrelatedness of events over long periods makes it impractical to discuss any notion of expected utility! Using the decision tree model is much more interesting because it facilitates taking into account the sequence of unfolding events (Raiffa 1968). However, the basic problem of assigning conditional probabilities to all conceivable scenarios remains. When the concept of expected utility becomes as complex as in the above example, Gilboa and Schmeidler (1995, 2000a) advocate the use of case-based reasoning instead.

In closing, it is useful to illustrate what paradoxical situations may arise if the model used to describe the decision problem is badly set. The following example shows a gambler attempting to use the theory of to decide on which horse to bet between two possible winners, *Lame runner* and *Ate the wrong stuff*. This example is presented in the shape of a question: which one of the two models presented in Table 1.2 is the correct one? (cf. Poundstone 1990).

Table 1.2. Comparison of the two models

	My horse wins	My horse loses
Bet on *Lame Runner*	50	–5
Bet on *Ate the Wrong Stuff*	45	–6

Model 1

	p Lame Runner wins	(1-p) Ate Wrong Stuff wins
Bet on *Lame Runner*	50	– 5
Bet on *Ate the Wrong Stuff*	– 6	45

Model 2

In the first model, *Lame Runner* is always the good choice because it is always on top. In the second model, the correct bet depends on the probability of a win for either horse and it is a better bet to pick *Ate the wrong stuff* as soon as the probability of it winning the race is above 50/106. Thus, in the first model, actions and events are incorrectly linked, whereas the second model is the correct one.

This example of drastically incorrect modelling shows the theoretical importance of Savage's formal framework for understanding decision making, quite apart from any consideration of expected utility.

1.2.2 Criticisms of Expected Utility

An important component in the debate around Savage's work centres on the way that the probability of occurrence of events is measured. Specific probabilities can of course be assigned to each event, but alternatively, it is also possible to assign a fuzzy measure of probability (see Dubois and Prade 1985; Bouchon-Meunier and Nguyen 1996; Bouchon-Meunier and Marsala 2003). Discriminating between events based on the likelihood of their occurrence is indeed quite tricky. For recurring events, it may be possible to measure their frequency of occurrence over time and to derive probabilities from this data. This would apply for instance to a computation of the probability that a regular train will be on time on a particular day. This is a *recurrent probability.*

This situation can be found in medicine for instance, where it is possible to derive statistics for typical pathologies within specific populations. On the other hand, it is of no use in the case where a manager attempts to predict the price of crude oil in a 6 month forward frame. In this case, probabilities do not apply in a rigorous sense. Savage's contention is that even when there is no way to estimate probabilities, an internally coherent decision making process will automatically imply a de facto assessment of the probability of key events. In other words, the very fact that one is able to properly select one action amongst others reveals one's inner perception of the probabilities at play.

Specific criticisms have also been levelled at Savage's work. Allais (1953) criticised the *sure thing principle* (Savage 1954) because Savage's vision of independence means that the utility function is linear with regards to the probabilities which, although required for the mathematical coherence of the model, is unlikely to be true in practice (at least not all the time). A second criticism centred on the axiomatic aspect of Savage's work refers to the principle of coherence, in situations where the decision maker ignores certain types of actions because they simply aren't "on his radar", and also ignores events that don't really have an impact on the decisions made. Finally, Savage was also criticised because the probabilities described in his work may make good theoretical sense, but mean nothing to real life decision makers. The notion that the decision maker can express the probabilities pertaining to all future events and that he or she can then maximise their expected utility is not realistic. The probabilities assigned by a decision maker can only ever be a priori because they do not follow from observation and subjective because they do not rest on any specific knowledge of future events. Certain researchers have indeed likened such probabilities to guess work lacking any objectivity (de Finetti 1937; Nau 2001). This then amounts to trying to model uncertainty with non-probabilistic models – for instance by using a maximisation principle (e.g.: where the decision is argument of $\mathrm{Max}_{\mathrm{A}}\ \mathrm{Min}_{\mathrm{E}}\ \mathrm{U}(a, e)$). The most sophisticated of these models also consider the influence of the worst possible results as in Jaffray (1988) and Essid (1997).

Even though the concern that the a priori probabilities assigned by managers are very subjective, is a valid criticism of the theory, it is always useful to remember that, in practice, this never prevented managers from making decisions! The observation of actual decision makers in real situations illustrates the two different paths that are typically followed in business: (1) find experts that are supposed to be able to provide reasonable probabilities and (2) forget about pure rationality and make *reasonable* decisions. This case broadly corresponds to Simon's notion of *Limited Rationality.*

To conclude on the work of Savage, it is worth noting that the critique of the role of probabilities can also be levelled at the role of the utility function. In the end, the decision that a manager should take in order to maximise expected utility is dependent on the chosen utility function and this is a fundamental problem from both theoretical and empirical standpoints. Knight's observations (1921, p. 230) on the confusion between risk and uncertainty is relevant here since he defined the former as relating to "the logic of probability" and the latter as "the problem of intuitive estimation". There is scope for applying expected utility theory in situations of risk when probabilities may be assigned, however arrived at (a priori or statistical). Where uncertainty prevails, any data that exists do not lend themselves to statistical analysis and "Business decisions, for example, deal with situations which are far too unique, generally speaking, for any sort of statistical tabulation to have any value for guidance. The conception of an objectively measurable probability or chance is simply inapplicable" (Knight 1921, p. 231).

1.2.3 Bounded Rationality

Based on his observations on the way in which the municipal decision makers of his town of Milwaukee made their decisions, Simon came to realise early on the distance that there was between managerial practice and the model of expected utility. Following this initial experience, he devoted most of his scientific career to trying to understand human decision making (Simon 1991).

He understood that, if the model of expected utility does not offer a complete explanation of human decision making, i.e. where uncertainty is of interest, the Taylorian vision of Dewey (as quoted in Simon 1977) is not much more relevant:

- What is the problem?
- What are the possible actions?
- Which one is the best?

This simplistic vision of the decision problem is hardly operational insofar as:

- "Unfortunately, problems do not come to the administrators carefully wrapped in bundles with the value elements and the factual elements neatly sorted" (Simon 1997); the environment of the decision is primarily ambiguous and depend on the personal interpretation of the decision maker (March and Olsen 1976; Boland 1979);
- Possible actions are not given but must be built from experience (see Keeney 1992; Roy 2000).
- The selection of the best course of action rests on the proper identification of the criterion for choice, which brings us back to our criticism of Savage's work or to multicriterion decision making (see Sect. 1.2.4).

Based on these observations, Simon insists on the diachronic aspect of the decision-making process and introduces his famous normative model of decision making stages, which from the initial three will become four (Simon 1977). Thus, Simon initially presented decision making as comprising three stages:

1. The identification of all the possible actions (or alternatives)
2. The determination of the consequences of all possible actions
3. The evaluation of the consequences of each possible action

Compared to Dewey's three questions, Simon's contribution is obvious. His focus is on the processes and he does not say: "what are the possible actions", but "we must find them all" (difficult question!). Let us note in passing that this presentation also has the merit to avoid the hollow question of "which is the best action". Thereafter, Simon adds several other aspects to the various phases of his decision making process, in particular with regard to problem representation, the way of posing the problem (or "setting the agenda") and the search for information. This leads to his seminal work on the four phases (Simon 1977):

1. Intelligence
2. Design
3. Choice
4. Review

The role of information is fundamental in the first two phases, for one chooses only among the actions which one identified and was able to document. Thus, as Simon indicated: information constrains the decision. Notwithstanding the criticisms levelled at his presentation of the decision process, Simon was perfectly conscious of the connections between the various phases and he provided examples of iterations between phases; even stating that each phase can be recursively regarded as a decision in itself (Simon 1977, p. 43). But undoubtedly the most significant contribution of this seminal normative model is that post-Simon, it has become more difficult to reduce the decision to the moment of the choice: "All the images falsify decision by focusing one the final moment" (Simon 1977, p. 40). This change of attitude will kill off a certain vision of the decision as mythology or epic (Julius Cesar crossing the Rubicon or De Gaulle launching the Concorde) to bring it back in the domain of management and a more scientific and systematic observation of its reality.

Finally, Simon was well aware of the fact that the decision, once taken, must still be implemented: "In the foregoing discussion I have ignored the fourth phase of decision making: the task of carrying out decisions. I shall merely observe by the way that seeing that decisions are executed is again decision-making activity" (Simon 1977, p. 43). He added (p. 44): "Executing policy, then, is indistinguishable from making more detailed policy". In the end, actions and decisions are inseparable for Simon and execution is merely a progression towards increasingly small decisions that can be readily implemented. This fundamental idea has yet to be exploited in management.

The framework defined by Simon makes it possible to connect decision and information but it is not rich enough in terms of understanding choice and analysing the role of future events. It is precisely at the core of the debate on the cognitive limits of human decision makers and their incapacity to predict events far in the future, which is necessary to apply the model of Savage. In other words the limitations of the brain and the nature of business decisions make it impossible to face the combinatory explosion of all the possible scenarios (Pomerol 2001). This led Simon to ask some awkward questions such as: how can a decision maker evaluate all the consequences of an action and compare them between them? We still don't have answers to these questions.

Simon had an interesting vision of the knowledge of the decision maker and his or her capacity to evaluate consequences (Simon 1997, p. 85). The problem of evaluation of the consequences of an action is central in any decision-making process. In Savage's work, the evaluation of the consequences supposes the knowledge of all the future events with their probabilities. In theory, it may be enough to maximize a function of utility for a set of choices, but the difficulty is to determine what is, in practice, the role of reason when there are neither clear choices, nor a complete utility function and managers operate with a minimal knowledge of future events.

In his book "Administrative Behaviour" Simon admits that the question we asked in the previous paragraph, in particular, about the evaluation of the consequences in uncertain situations is not solvable by a human mind in the terms of the expected utility model. Simon calls this "absolute" rationality which would require that one chooses, following the model proposed by Dewey, the best possible action (i.e. an optimised choice) having evaluated all possible consequences going 100 years into the future. According to Simon, this *substantive* rationality, as he later called it, is a practical failure because (Simon 1997, p. 93–94):

- Rationality requires a complete knowledge and a total anticipation of the consequences of all choices. In practice knowledge on the consequences is always partial especially in uncertain or ambiguous situations; This question of exhaustiveness is also central in Janis and Mann (1977) and Klein (2002)
- Consequences are a matter of speculations and the mind must fill in the blanks in assigning values to them
- Rationality requires choosing among all the possible actions that have been identified (March and Simon 1993, p. 159). In reality, only a small number of possible actions come to mind
- The decision maker does not hold a complete set of preferences for all possible consequences, i.e. he or she does not have complete utility function (March and Simon 1993, p. 159). There are therefore difficulties inherent in the ranking and comparing of the alternatives (Janis and Mann 1977).

The core criticism levelled by Simon boils down to the fact that, except in very simple cases, using subjective expected utility (SEU) in a correct way is simply impossible. Indeed, his criticisms presented above are aimed squarely at the implicit assumption of the model of expected utility. He said: "When these assumptions are stated explicitly, it becomes obvious that the SEU theory has never been applied and never can be applied – with or without the largest computers – in the real world" (Simon 1983, p. 14). The volume of knowledge necessary to apply the model justifies that Simon should call it the *Olympian* model (Simon 1983, p. 19). In his work, Simon will endeavour to replace these *Olympian* assumptions with realistic assumptions. In 1955, these assumptions will then become the basis of bounded rationality. These can be summarized as follows:

- It is impossible to assign probabilities to all the events and even quite simply to enumerate all the possible events with their permutations.
- The preferences of the decision maker are not rational insofar as there is no possible maximization of a utility function. In fact, they are multi-criterion

and variable, which means it is impossible to have a complete utility function for the choice made.

- Decisions and their consequences are spread out in time and, in organizations, form a temporal process in which all sub-decisions are not independent from other sub-decisions, but can be made at different times and levels based on evolving criteria. In addition, preferences, actions and goals cannot normally be readily separated ("closely related to the idea that actions generate their goals is the fact that action is itself an important goal in the lives of many people" (March and Simon 1993, p. 15)); The articulation of the sub decisions as described above rules out any form of overall optimization (Simon 1983, p. 18).
- Information is fundamental and conditions each decision. This is perfectly illustrated by the small number of actions which an individual is able to study seriously. The limited attention of managers further constraints and limits the analysis of the problems facing them and conditions subsequent decisions. Attention is a rare resource and it tends to be concentrated on the most salient problems.

This means that, since we cannot have complete knowledge of the world, we, as human decision makers must aim at making sub-optimal or satisfactory decisions, which Simon labelled "satisficing". In practice, the decision-making process stops as soon as the decision maker finds a solution which gives satisfaction taking into account the most plausible scenario, and is also unlikely to turn out to be catastrophic. Simon (1984, p. 594) evokes explicitly how "satisficing" operates. He explains that an action is satisfactory as long as it reaches or exceeds a certain level of aspiration for the criteria considered by the decision maker (March and Simon 1993, p. 161). It must also be noted that the level of aspiration evolves during the intelligence phase and is interpreted at a local level depending upon the difficulties of reaching it (Selten 2002). The concept of "satisficing" tends to become increasingly important in Simon's work after 1960 such as Simon (1983). The limited rationality of 1955 is gradually replaced by the "bounded rationality" (Simon 1972). This "bounded rationality" is more and more frequently presented in algorithmic form as was already implicit in 1955 in the form of a "satisficing rule". The algorithmic aspect stresses the sequential and heuristic aspects of decision-making processes. Thus, following Gigerenzer (2002) it is possible to summarize the notion of bounded rationality with a number of fast, rough and robust rules: (1) for the intelligence phase, (2) to stop searching for information and (3) to make a choice (Gigerenzer 2002). This vision justifies the use of the term *procedural rationality* (Simon 1976) which Simon opposed thereafter to substantive rationality. This evolution in Simon's thinking is accompanied by an increasing interest in artificial intelligence ("Alternatives of action and consequences of action are discovered sequentially through search processes" (March and Simon 1993, p. 191)). The heuristic process involved is characterised by the use of procedural rationality, because rationality is used in the search for information, while at the next stage, the manager's thought process or "problem solving" is characterised by substantive rationality (March and Simon 1993, p. 200). In searching for information, managers follow a form of procedural rationality which obeys a program just like a heuristic search. The criterion used to interrupt the search is the satisfaction of the decision maker when a "satisficing" level is achieved taking into consideration his or her aspirations.

The fourth limitation of rationality in our above list, is critical because it presents a dual aspect. Firstly, there is the informational aspect – i.e.: that the quantity of information which an individual can process is limited. In the "information age" where we are plunged, the gap between the information potentially available and what a decision maker can apprehend is widening (it is even truer with the Web). Simon (1955) explained: "Broadly stated, the task is to replaces the total rationality of Economic Man with a kind of rational behaviour that is compatible with the access to information and the computational capacities that are actually possessed by organisms, including man, in the kinds of environments in which such organisms exist". This first aspect leads to a second idea: that the cognitive resources are also limited (Bell et al. 1988). In fact, one already finds in Simon's "administrative behaviour" the first reflections on the role of attention, information and the stress in the decision process (in the chapters devoted to psychology). These considerations will lead Simon to the problem of cognitive load in decision making. He describes attention as a rare resource (especially in view of the limited cognitive capacities of human beings) which plays an important part in the decision process. This topic is pursued in his book with March ("... the ways in which attention is allocated is critical to understanding decision" (March and Simon 1993, p. 4)) and becomes one of the key elements in the garbage can model (Cohen et al. 1972).

As Simon's thinking evolves, cognitive limitations gradually became a major element of limited rationality by reference to the brain as a system for symbolic processing. "In its simplest form, the theory of limited rationality is a theory of "how to live" in an infinite world, while having only very modest means of computation; means which do not depend on the size of the real world, but only of the local environment and what you can do there" (Simon 1984, p. 595). Simon's contention is that managers must make do with their capacities which rules out the exhaustive study of all possible actions and their consequences. Thereafter Simon will often oppose the procedural rationality which is the rationality whereby human beings seek to understand the consequences of actions with their limitations in information, in cognitive capacity and in attention, which is inherently a *satisficing* rationality[5] leading

[5] "The idea of limited, consequential rationality found in the book has become more or less standard in modern theories of decision-making, at least outside the hard core of orthodox neoclassical economic theory" (March and Simon, 1993, p. 9)

to satisfactory decisions, as opposed to the substantive rationality which is the preserve of the Gods and which is inherently an *optimizing* rationality.

The model of limited rationality is, according to Simon, a middle of the road model (Simon 1997, p. 331) half way between the point of view of some Economists who tended to believe in pure rationality but some of whom now examine alternative models to the maximization of utility and, on the other side, the point of view of those that the notion of rationality frightens and who argue that managers are purely reactive and intuitive in their behaviour (e.g.: case-based reasoning research). As we see it, bounded rationality was the first attempt to provide a scientific framework for the rigorous and meaningful study of real decisions made by real decision makers in real life organizations. This explains why the concept of limited rationality has had such an impact, even 50 years on.

1.2.4 Multi-Criterion Decision Making

Simon was one of the first researchers to express with a certain scientific authority that real life decisions are characterised by more or less contradictory criteria insofar this observation is one of the components of limited rationality. This observation had obviously already been made by real life decision makers and Benjamin Franklin suggested the "for and against" method where arguments for and against are cancelled out until one of the columns is empty (letter with Joseph Priestly, see Zionts 1992).

The concept of multi-criterion decision making is fundamentally human in the sense that everyone wants to "have their cake and eat it". This problem has of course no solution and yet people carry on making decisions (Keen 1977) unless they elect to stay in a non-decision making scenario (Judge 1997), which is, in itself, a form of decision making. The need to arbitrate between short term and long term is an excellent illustration of inevitable and sometimes painful multi-criterion choice. How can compromises be made? From the neurobiological point of view, we have seen in Sect. 1.1 that the ventromedial part of the frontal cortex is a key centre and that certain aberrant behaviours come from a failure to integrate available information, the dominance of short term gains and uncontrolled sensitivity to certain emotions.

As illustrated by Gilboa et Schmeilder's first example (the manager trying to hire a sales representative), multi-criterion decision places more emphasis on the description of the characteristics of the possible actions than on the events to come. In a certain manner it is better to spend time on a good evaluation of a potential action, rather than to endlessly consider highly uncertainty events. This is why the proponents of multi-criterion decision making appear somewhat indifferent to uncertainty: "Information versus Uncertainty" is indeed a recurring theme. By the same token, fast decisions are better than long studies of hypothetical events to come (Eisenhardt 1990), especially when decisions are not irreversible (Pomerol 1997a). That has been illustrated in experiments such as the "beer game" (Sterman 1989) and in empirical studies of real decisions with delayed feedback (Kleinmuntz 1985, 1993).

Fundamentally, human actors don't like the tension inherent in multi-criterion choices (Kottemann and Davis 1991; Berthoz 2003, p. 286) and very often will seek to rationalize their choice either by the search for dominance (Montgomery 1983, 1987), or by reasoning by analogy, but almost never by having recourse to aggregation, which seems to be an effort to rationalise limited to the scientific community. Thus, the decision maker will often prefer to use heuristics and limited rationality, to proceed by trial and error using interactive methods (see Pomerol and Barba-Romero 1993) and local adaptations fitting their levels of aspiration (Lévine and Pomerol 1986; Selten 2002). These decision making traits are exacerbated in certain models, such as Klein's (1993) recognition-primed decision making where only one scenario is considered in great detail and its implementation monitored against the elements that emerged from a rapid simulation carried out by the decision maker in his or her mind.

1.2.5 Other Models

We considered the problems arising from the use of the probabilities and the concept of expected utility in relation to Savage's model. We have also shown how it is possible to bypass these problems by adopting alternative models, such as MaxMin. In practice, sensitivity to the worst result is a phenomenon well attested (March and Shapira 1987; Tversky and Wakker 1995). Tversky and Simonson (1993) have even coined the term "extremeness aversion" to describe it.

Researchers have tried to construct models that take into account the probabilities and the aversion for overwhelming losses (e.g.: Cohen and Jaffray 1988; Jaffray 1988; Rubinstein 1988; Leland 1994). A more complete attempt consists in taking into account the difference in value between the results versus the difference between their probability of occurring (Shafir et al. 1993). Such models try to recreate a hybrid selection criterion by introducing the aversion to strong losses or great differences in profits. The issues arising from the existence of events with very small probabilities are important ones, because they are one of the main sources of error of judgment in human decision making (March and Shapira 1987; Morel 2002). The use of belief functions as in Dempster (1967) and Shafer (1976) also allows for a mix of beliefs on the probability of future events and some degree of ignorance. In Smets' (1990) model, the belief functions are transformed into probabilities at the time the decision is made in a transformation process known as pignistic transformation (see Dubois et al. 1996).

The alternative perception of probabilities as illusory precision is also a legitimate one and Dubois and Prade (1985) have suggested replacing them with possibilities, which are sub-additive measurements (i.e. the measurement of two independent events can be lower than the sum of the measurements of each event). It is then sufficient to rank the events from the most probable to the least probable and only the rank of each event in the list counts. It is then possible to use a Choquet integral to integrate the results and obtain a probabilistic expected utility as well as other decision criteria within a probabilistic axiomatic framework (Dubois and Prade 1995; Dubois et al. 2001). The result provides a *qualitative* decision because only the relative plausibility of the events is taken into account without an absolute measure of their probability of occurrence intervening. Dubois et al. (2003) provide a good synthesis of the various models and criteria which rest on weaker measures than actual probabilities. Much recent research has shown that models as coherent as that of Savage have been proposed in this way (Dubois et al. 2002). In some cases however, such models lead to an over-focus on the most plausible events in setting up the decision (Dubois et al. 2002, 2003).

1.3 Decision Making, Pattern Recognition and Look Ahead

1.3.1 Diagnosis and Decision

We have already stated that it is impossible to describe human decision making without considering the role of future events. By contrast, a deer's sudden decision to run away is a mere reaction to a stimulus. This flight reaction is built into the animal's genes and does not entail a representation of the future. Naturally, humans may display such automatic behaviours in some cases, such as ducking when an object is thrown in one's direction. In the domain of *reasoning*, (i.e. when the decision maker has enough time to generate a projection of future events in her or her mind), it is useful to distinguish between two key phases: *diagnosis* and *look-ahead*. It is, of course, not always easy to separate these two but, from an engineer's point of view, it facilitates the design of systems aimed at supporting the process of decision making. In Fig. 1.1, we have sketched out what may be regarded as a realistic human decision process, tracking the main components of decision reasoning. In Fig. 1.1 we have drawn a line from the preference box to the actions because many consider that it is possible, to some extent, to define the actions according to preferences. First define what you want, then design the actions that will get you there! This is expressed in current research originated mainly by Keeney (1988, 1992), about value-driven thinking. Here attention is drawn to the fact that the action (or alternative) set is not a given and can be changed during the process of reasoning.

It has been often observed that many real decision makers are overconstrained in their perception of the alternative set and study just a small subset of the possible alternatives. Classical decision theory assumes that the actions are known, even though it has for long been recognised that the design of the actions itself is an important step in the decision process (Simon 1977).

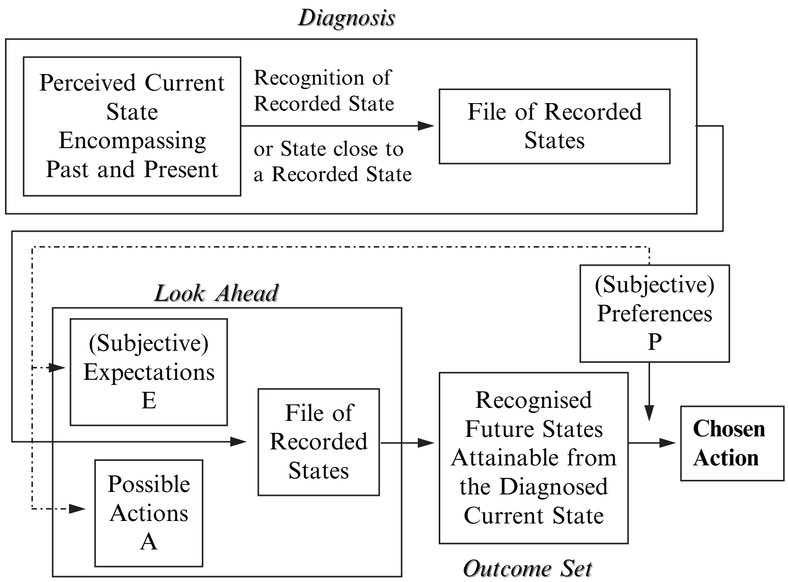


Fig. 1.1. The decision process (adapted from Pomerol 1997a)

In some cases, it is also defensible to draw a line from the preferences to the expectations box. This may be regarded as a psychological bias because it means that the future is considered in terms of the preferences. This probably frequent situation should be avoided in rational decision making, as should the inverse situation where the preferences are influenced by expectations. The latter can be regarded as a kind of framing effect (see e.g. Tversky and Kahneman 1983, 1988 and Humphreys and Berkeley 1985, for a discussion). Indeed, rationally, preferences should be independent from expectations.

Also, the subjects' preferences may influence the diagnosis process and the file of the recorded states (i.e.: the memory). Numerous psychological biases are observed in this domain (von Winterfeldt and Edwards 1986; Bell et al. 1988). Another simplification in Fig. 1.1 is that the decision process may appear "linear". This is not the case and many backtracks can occur, especially when the subject becomes aware that the attainable future states are not satisfactory. Moreover, in many actual organisational settings, due to feedback phenomenon, it is not always possible to distinguish an event from an outcome. For example, in an oligopolistic market, are the rise and falls of a price an event (uncontrolled and uninfluenced) or an outcome? In many cases, the decision makers and the modellers do not know, on the one hand, where to set the limit and the time horizon of the model because, depending on the level of analysis, any significant decision may have far-reaching consequences (see Berkeley and Humphreys 1982, for a discussion about the small world assumption), and on the other hand, the line between events and outcomes is rather thin and vague in terms of human agency.

In Fig. 1.1, the diagnosis phase consists in recognizing the current state of the world, i.e. the past and the present. In the next phase, the decision maker must anticipate the consequences of potential decisions, based on his or her perception of the future, it is the projection phase. This is the stage that best distinguishes human decision making from animal decisions. Even though it is logical to imagine that the appearance of an increasingly present projection phase in our decision making occurred gradually during our evolution, there is a stage where this decision making phase became the most important and paleobiology does not allow for a conclusion regarding which of our ancestors had or did not have access to such capability. The evolution also explains why in human behaviour certain situations still involve decisions that are either automatic, or based on the recognition of patterns.

We have argued that decisions made directly on the basis of the recognition of a state of the world, i.e. a diagnosis calling for a standard reaction, was a frequent and even sometimes rational process, in particular for continuous types of decisions such as in industrial process control (Pomerol 1997a). Expert systems were based on such concept: a good diagnosis leads to the decision, whether one represents the states of the world in the form of rules as in the expert systems or in the form of cases (see Riesbeck and Schank 1989 and Kolodner 1993 for an introduction to case-based decision making). The phase of diagnosis consists in recognizing a state of the world. In cases where an exhaustive list of the "diagnosable states" is present, together with a list of decisions such that a one-to-one relation can be built between the two, decision tables can be used as the decision taking device (see Pomerol 1997a).

The situation is often more complicated in particular when the diagnosis does not make it possible to identify a case already recorded in the memory. We will examine the model of Gilboa and Schmeidler (1995, 2000a) which tackles this question of the recognition when not all the "recognizable" states are present in the memory of the decision maker.

1.3.2 Case-Based Reasoning

The principle of case-based decision making is simple. It assumes that there is a set of decisional cases in the mind of the decision maker and that these cases represent all the experience of the "decisional system". Faced with a new situation, the decision maker recognizes one of the cases already encountered and initiates the decision adapted to this case (decision which has also been stored). In the simple case of the decision table scenario, the difficulties which arise are purely "representational", i.e. it is necessary to have an advanced language or a representation scheme which makes it possible to capture the richness of each case and authorizes a rapid pattern matching. These present key questions for ARTIFICIAL INTELLIGENCE researchers which are dealt with in the chapters of this book.

In reality, Case-based reasoning is not only about pattern matching insofar as, as the proponents of CBR have rightly claimed - the learning dimension of

CBR systems is the most important one. For instance, the set of cases must be able to grow to encapsulate any newly encountered case which does not fit existing cases. The system must also be able to deal efficiently with any unrecognisable case that is encountered. The issue of similarity between cases becomes a critical one, with the system having to properly assess the distance between any new case and one or several existing cases. Gilboa and Schmeidler (1995, 2000a) proposed a framework to formalise the relationship between case based reasoning and case-based decision making. They propose that each case is a triplet (p, a, r) where p $\in$ P (the set containing all problems), a $\in$ A (the set of possible actions) and r $\in$ R (the set containing all results). Case-based reasoning is concerned with the problems and how to classify them in comparison with each other. Gilboa and Schmeidler defined a similarity function between problems:

$$\mathrm{S} \;:\; \mathrm{P}^2 \rightarrow [0,1].$$

This function gives the distance between two problems. The decision maker can also use a utility function on the outcome:

$$\mathrm{U} = \mathrm{R} \rightarrow \mathrm{IR}.$$

Let M be the set containing all cases stored in memory: the relevance of a given action for a given problem is expressed as:

$$U_p(\mathrm{a}) = \sum_{(q,a,r)\in M} s(p,q)u(r).$$

In other words, for a given a and p, all problems q in the memory that satisfy $(q,\ a,\ r) \in M$ taking into account their distance to p (i.e. s $(p,\ q)$) which increases when q gets very close to p. It is then logical to select the action a that maximises U_p (a). Gilboa and Schmeidler (1995) also provide axioms which show the coherence of their model. As in Savage's model, a coherent choice of an action yields a measure of the distance between the problems (instead of the probabilities of the events in Savage's model). This similarity between these two types of model proves – indeed it is one of the great weaknesses of this type of model – that the reasoning on future events (i.e. uncertainty) is contained in the similarity function built in the model. In Gilboa and Schmeidler (2000a), the model is extended to the similarity between the pairs (problem, action) and the triplets (problem, action, result). By contrast with Savage's work, case-based reasoning (as in the previous paragraph) has a significant advantage that instead of knowing all the states of nature and the consequences of the various possible actions, it is enough to have a memory of all previous cases. It remains to be considered whether the set of previous cases has pertinence in understanding future events. It is therefore of great interest that the set of recorded actions is allowed to grow richer by the introduction of new cases, but also by the refinement of the similarity function, as the model is used.

In a recent work Gilboa and Schmeidler (2000b) proposed an axiomatic model to derive probabilities on the basis of a set of recorded cases. The principal element of appreciation is the number of occurrence of the cases, a high number of occurrences resulting in a higher associated subjective probability. It is another way to model the availability, reinforcement (Anderson 1983, 1995) and representation phenomena (Tversky and Kahneman 1982a, b).

1.3.3 Recognition Primed Decision Making

In considering the specificities of human decision making and the importance of the projection phase, especially when the patterns faced by the decision maker is not an exact match for any previously experienced situation, it is useful to discuss in some detail the work of Gary Klein (see Klein 1993) and his associates on recognition primed decision making (RPD) and naturalistic decision making (NDM). NDM is at the same time a body of research on human decision making and a methodological orientation which has focused on the study of certain cognitive functions that emerge in natural settings, often in decision making situations that involve severe time pressures and/or life and death decisions. At the outset, Klein and the adopters of his ideas studied fire fighters, emergency room nurses and paramedical staff with the view to getting direct observations on how this extreme decision making can occur. Their observations reveal that, contrary to the predictions of normative models, these individuals do not design alternative solutions that they compare with one another, but use their diagnosis of the situation to construct very quickly a best case solution, the execution of which they then simulate in their minds to see if it is a good fit for what they imagine is happening: this is what is termed recognition primed decision making. This is radical, because it proposes that RPD is about selecting one solution and running a quick simulation in one's mind to test its robustness. In the following phase, when the solution is implemented, Klein's decision makers use their simulation to validate whether the situation responds in the way they expect and take further decisions as they see an unexpected course of event developing, such as this fire fighter getting his team out of the building seconds before the ceiling collapses on them because he identified that the fire should have been reduced by their attacks and he understand that the fact that it is not abating means a faulty diagnosis (i.e.: the incorrect identification of the location of the fire in the building).

These observations are very interesting, because they illustrate well the importance of experience and why more experienced decision makers are less likely to get themselves and their teams in trouble. As illustrated in Fig. 1.1 in Sect. 1.3.3, the size and variety of the file of recorded states, determines both the likelihood of an accurate diagnosis and the likelihood of an effective solution being implemented. It also allows for a more detailed monitoring of the implementation progress. Thus, RPD suggests a two stage decision process: (1) pattern recognition and (2) mental simulation. Klein concluded from his observations that human beings are active interpreters of everything they see around them and their experience cannot be deconstructed into the kinds of rules that will fit into expert systems. He also noted that experienced decision makers see a different world than novice decision makers see and that what they see tells them what to do. Ultimately, it remains to be seen whether RPD is a specific form of decision making that applies to certain individuals in certain situations or whether it can be considered as a broadly applicable alternative model of human decision making.

1.4 Recognition, Reasoning, Decision-Making Support and the Use of Scenarios in Decision Making

For anyone interested in human decision making, it is enough to read the recent work published in neurobiology to be convinced of complexity of the neuro-processes involved (see Damasio 1994; Damasio et al. 1996; Berthoz 2003) and it is important to repeat that conscious and deliberated reasoning is not the only domain of research that must be considered: recognition, as we have illustrated in the previous sections, is also a critical aspect of decision making. Indeed, even in animals, pattern recognition is moderated by context and training (Berthoz 2003). This has led researchers to conclude that a sound alternative to trying to model such a difficult activity was "to leave the human actor in the loop" and to focus on interactive decision support systems (DSS). DSS aim at assisting rather than replacing, the human decision maker by providing rational models to support his or her reasoning abilities and by extracting relevant patterns in vast volumes of overabundant information to support his or her recognition abilities. There is a vast literature on these systems, including, Keen and Scott Morton (1978), Bonczek et al. (1981), Sprague and Carlsson (1982), Lévine and Pomerol (1989), Burstein (2001), Humphreys and Brézillon (2001), Adam et al. (2002), Mora et al. (2002), Adam et al. (2003). In final analysis, decision making, when the decision maker has time to consider alternatives, boils down to the ability to build representations of the (uncertain) future and to project oneself in it. Consequently, supporting decision-making is initially concerned with the construction of scenarios and the amplification of this specifically human aptitude to project in the future in a conscious way.

It is useful to note that despite genuine advances in practice, with countless applications developed and implemented with success in industry, we still lack a strong theoretical basis to integrate the very disparate systems we are aware of into a coherent whole. One promising direction of research consists in regarding the use of DSS applications as performing a heuristic search for a solution. (Bonczek et al. 1981; Lévine and Pomerol 1989, 1995). The DSS then, facilitates this heuristic search by helping the user to explore the future ("what if analysis"). This exploration must be done at two levels: the level of the data and the level of the models (Pomerol and Adam 2003). It is the need for this dual degree of freedom which makes such applications as spreadsheets so popular and so effective (Pomerol et al. 2002, 2003). Obviously the heuristic search stops when a satisfactory solution ("satisficing") is found. This is a perfect illustration of Simon's bounded rationality in action.

Given the unlimited number of possible future states of the world, the human decision maker, helped by his or her decision support artefacts, will develop scenarios, a small number compared to all those possible (Pomerol 2001). These scenarios will be projected against a given timeframe depending on the context of the decision, but which can be quite long for strategic problems, hence the need for a DSS, because the combinatory explosion can grow well over the ability of the human mind, even for a small number of scenarios.

The use of scenarios appears to have been both the most common and the surest way to explore the future. In its most formalized form, it has given rise to the use of decision trees (Raiffa 1968; von Winterfeldt and Edwards 1986; Shafer 1996), decision tables, or other graphical methods described elsewhere. By assigning conditional probabilities to the various successive events that make up the scenario, decision makers enter a mode of reasoning referred to as backward folding, which makes it possible to rigorously determine the scenario with the best expected utility. It is interesting to note that, due to the opposition between the short term and the long term, the best scenario is practically never the chaining of the best scenarios at the intermediate stages.

Many other graphic methods have been derived from decision trees, for instance, towards decreasing the need for independent probabilities between all the events (networks of influence, Bayesian networks, see Oliver and Smith 1990; Shenoy 1994) and so have various qualitative methods (Oliver and Smith 1990), some of which use no probabilities to represent the context of the decision (Brézillon et al. 2002; Pomerol et al. 2002).

One of the important aspects of the practice of decision making which is corroborated by many empirical observations (Pomerol et al. 1995; Brézillon et al. 2002) relates to the simplification of the scenarios in human reasoning. This involves the use of actions that are considered to be robust in the face of large series of events belonging to a common temporal threat, and which make it possible to eliminate a number of problems at the same time (Pomerol 2001). This leads to successions of standard decisions which are quite common in practice.

Another way to reduce the combinatory explosion already discussed consists in delaying as many decisions as possible until the end of the scenario. This has been called action postponement in Pomerol (2001). This can be interpreted as the continuation of the search for information before making a decision (when information is available), or as an illustration of the old saying about not keeping all your eggs in one basket, in this case by delaying the decision point until after as many uncertain elements as possible have disappeared. In dynamic programming, this capacity to keep as many options open for as long as possible is referred to as flexibility (Rosenhead 2001). These very pragmatic ways of thinking, even though they do not add up to substantive rationality, are perfectly "rational" and fit well with the idea of using scenarios for reasoning.

In this perspective, the question which decision support systems must help answer concerns the choice of the most useful scenarios. In many cases, this choice of scenario is a multi-criterion problem, in the sense that the decision maker must make conscious trade offs between possible criteria for success. Although it is uncertain exactly how the human mind processes these cases, neurobiologists (Berthoz 2003) claim that the mind relies more on the elimination of potential solutions than by choice. In other words, through a complicated physiological process bringing into play many parts of the brain, a dominant solution ends up inhibiting all other possible solutions. This is reminiscent of the phenomenon of search for predominance described in Montgomery (1983, 1987) and of the empirical results obtained in Psychology in experiments where an individual convinces themselves a posteriori that they bought the best car or bet on the right horse given existing constraints (Festinger 1957). It can be hypothesised that there are thresholds in the discharge of our brains' neurons which result in a "winner takes all" phenomenon. This type of phenomenon is measurable in multi-criterion decision making but, it is essentially hidden in the stage where weights or relative importance are allocated to each criterion. As Keen (1977) pointed out: the most interesting in human decision making is that even when there is no solution in theory, we go on making decisions in practice – judgement is exercised.

In the selection of scenarios, robustness plays an important part. Robustness may be understood to relate to events, to data and to the parameters built into the models used (see Roy 1998, 2002; Vincke 1999). In considering robustness as it relates to events and their probability of occurrence (or any other measurement as discussed earlier), one is reminded of the comments made about Savage's framework: it is extremely complex to abandon the maximisation criterion, robust "against any move of nature" and to try to establish which events are negligible or not (Lehmann 1996; Monnet et al. 1998). It is all the more difficult given the weakness of the human mind in appreciating small probabilities. This cognitive bias is found in many reported accidents, such as the loss of the space Shuttle Challenger. The designers of the boosters used to propel the craft during take-off were confident of warm weather, based on historical data showing only one or two days of cold per century in Florida. Unfortunately, this particular launch took place during a cold spell which led to the disastrous explosion (Morel 2002). It is noteworthy that the expected rate of failure for such spacecrafts was assumed to be 1% by its designers and 1 per 100,000 launches by the managers of the Space Shuttle project. This difference in probability leads to radically different behaviours: one may perceive a 1% scenario worth considering, whereas a 1 in 100,000 chance scenario can be neglected.[6] Sect. 1.5 concentrates on the limitations of the human mind, such as this problem with assessing small probabilities, and other reported cognitive biases.

1.5 Cognitive Biases

There are unavoidable obstacles which defeat all efforts at rationality in human decision making. The first one concerns small probabilities: should a low probability of disaster (such as total bankruptcy) automatically rule out a possible action? Should one leave the car at home when there are high winds? Decision makers may either take a very pessimistic decision criterion and remain in bed all day, or treat these exceptional situations as exceptions and display basic logical incoherence (Dubois et al. 2003). The sure thing principle is another obstacle because it imposes a "rationality" that nobody accepts: there are good reasons to buy the fastest car even though its petrol consumption is greater than that of other models if you have a large budget for your purchase. On the other hand, if you budget is limited, you are likely to look for an economical model, even if it goes slower. The assumption of linearity of the preferences as they relate to the probabilities (or to the weights in multicriterion decision making) is unavoidable. However, it is purely mathematical and is not particularly rational because it is quite conceivable to change one's mind in relation to one's preferences depending upon the level of satisfaction that can be obtained. The ignorance of other axioms not discuss so far, can also yield severe inconsistencies in human decision making. It is the case for the axiom known as the "irrelevant alternatives" axiom (see Pomerol and Barba-Romero 1993) which, when it is not satisfied, leads to such paradoxes as that exploited in his time by Talleyrand where individuals can be forced to make one particular choice regardless of their own preferences (Woolsey 1991). In this case, very bad or very expensive choices are introduced in order to push decision makers towards a particular choice, for instance the median choice!

These facts are well attested in laboratory experiments, as is the violation of the principle of independence. Following Allais' foundational critic, many experiments have confirmed this phenomenon, notably the work of Kahneman and Tversky. These results are presented in Sect. 1.5.1 as they relate to some of the problems discussed thus far (see also Kahneman et al. 1982; Kahneman and Tversky 2000).

We will not spend undue time on the emotional aspects of certain decisions, such as the "frame effect", or on the effects which the presentation of

[6] "There is a tendency in our planning to confuse the unfamiliar with the improbable. The contingency we have not considered looks strange; what looks strange is thought improbable; what is improbable need not to be considered seriously", T. C. Schelling (1962, p. vii)

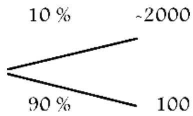
the context of a decision can have on decision makers (Tversky and Kahneman 1988; Slovic et al. 1988). Many experiments have revealed this effect identified a long time ago by Tversky and Kahmeman. Notably, Zickar and Highhouse (1998) have shown that the importance of this effect depended on each individual and Slovic et al. (2002) reported many examples of the sensitivity of human decision makers to the presentation of the facts of a decision. In brief, if one presents the same situation in term of possible death or in term of survivors one often manages to reverse the judgement of the majority of subjects. It is obviously purely irrational as are techniques that have been developed to manipulate public opinion using very small probabilities combined with the so-called principle of precaution (e.g.: invading a country because there is a possibility that it possesses weapons of mass destruction). In the following paragraphs, we will consider cognitive biases relating to probabilities, and those related to the anchoring effect and to the levels of aspiration of decision makers (Kahneman et al. 1982; von Winterfeldt and Edwards 1986; Bell et al. 1988; Kahneman and Tversky 2000).

1.5.1 Cognitive Biases Related to Probabilities

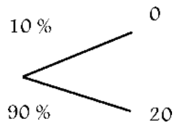
We already noted that small probabilities are not correctly apprehended by the human mind, in that they either are ignored (March and Shapira 1987), or over-estimated (Tversky and Wakker 1995). To tell the truth, between a probability of 10^{-3} and one of 10^{-6}, it is difficult to properly represent what the difference means and, without an emotional content, the mind has no point of reference. However, between catastrophic floods which occur on average every 3 years or every 3,000 years, there is a big difference for the inhabitants of an area. Experimentation has shown that 10^{-3} seems to be an important threshold for the perception of risk in human decision makers. Below 10^{-4} individuals tend to disregard the risk: it is the same probability as getting 12 or 13 consecutive heads when tossing a "regular" coin. Below 10^{-3}, the risk is accepted within certain limits if there is a perception that it can be somewhat controlled – e.g.: the decision maker thinks that if they really pay attention, they will get through safely (Perrow 1984, Chap. 9; McKenna 1993). For an average driver living in France and driving 20,000 km per annum, the risk of personal injury is 1/300 and the risk of a fatal accident is 1/4,300 (1997 statistics). The risk of a fatal accident which mountaineers face if they go out for a serious climb once a year also ranges between 1/500 and 1/1,000. For a "frequent flyer" travelling around 20,000 km per annum the risk of death is 10^{-5}, which is considered negligible. The example of road traffic is very interesting, because it shows that even with a non-negligible probability of serious accident, drivers are happy to undertake difficult journeys on "heavy" days (such as long week ends and holidays) with their families on board because they feel that being careful reduces the probability to within acceptable levels of exposure.

At the other end of the scale of probabilities, the effect of certainty is also well attested. The certainty of winning of smaller amount is always preferred to the possibility of winning a larger amount with a probability $1 - e$, or nothing with a probability of e, even when the expected utility is exactly the same. But in this case the common sense rationality at play is obvious: a bird in the hand is better than two in the bush! This takes us back to our previous discussion: if this behaviour appears rational for $e = 10^{-3}$, it is more difficult to justify it for $e = 10^{-6}$ but human nature is inherently risk averse when it comes to gains (see Kahneman and Tversky 2000, part three).

People don't like to lose and this has been amply demonstrated since work by Kahneman and Tversky (1979). Human behaviour faced with uncertainty, is not the same for profits or for losses. We have already noted the impact of possible large losses (e.g.: total bankruptcy) and the aversion they generate in human decision making (March and Shapira 1987; Cohen et al. 1987; Tversky and Simonson 1993; Tversky and Wakker 1995). By contrast, human decision makers are happier to take great risks in situations involving losses. Thus, human decision makes are risk takers when it comes to losses. In experiments, subjects were happy to face the risky odds in the figure below in order to avoid a sure loss of -10:



This means that subjects preferred an expected utility of -110 rather than a sure loss of -10! This type of behaviour may in part explain certain gambling addictions where individuals try to bail themselves out by taking increasingly greater risks. This can be contrasted with subjects preferring a sure gain of 10 to the situation proposed in the figure below, which has an expected utility of 18:



It seems that subjects' perception of the real utilities is somehow altered such that their utility curve looks like the one presented in Fig. 1.2. This preference reversal (PR) between gain and loss perception described by Kahneman and Tversky (1979) is a key notion for anyone hoping to understand human decision making.

We will not discuss here what happens when Bayesian rules are not respected or the incapacity of human decision makers to properly account for conditional probabilities. It is obvious that the human mind is not built like a calculator, so when it comes to computing conditional probabilities, its reliability breaks down. This is not really a weakness if one considers the

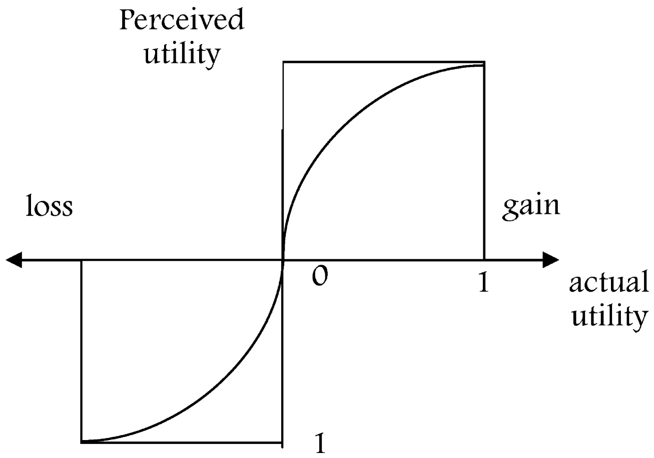


Fig. 1.2. Subjective perceptions of utility

intractable cognitive load involved as soon as 4 or 5 possible events must be considered with their conditional probabilities. This is where scientific reasoning and scientific tools must take over. Thus, in medical decisions (Grenier 1990), the last 50 years have seen unprecedented progress in terms of diagnosis, with the application of Bayesian and conditional probabilities. It is also worth noting that the issue of the coupling of events plays a significant part in the assessment of reliability of systems or machines and the risk of accident can be strongly underestimated if events are wrongly understood to be independent when they are not. Perrow (1984) demonstrates how tight coupling between very complex systems can lead to sequences of seemingly unconnected events which result in serious accidents.

The last effect that we would categorise as relating to probabilities is referred to as the illusion of risk control (Kahneman et al. 1982, part V; Slovic et al. 1982; March and Shapira 1987; Kahneman and Lovallo 1993; McKenna 1993; Barki et al. 1994). The notion of risk control is somewhat irrational and can be regarded as a pre-Savagian regression where a confusion is allowed between what the decision maker does and does not control. It amounts to refusing the principle of separation between actions and events and leads to paradoxes and incoherence as described in Sect. 1.2.1 with the betting example. The only reasonable notion when it comes to controlling risk, is that greater, more systematic information retrieval, and better forecasts, gives a better understanding of what may happen. For instance, using weather forecast for planning one's commute on foot or in a car reduces the risk of getting wet, and, in certain countries, weather experts give their forecasts in terms of probabilities. The search for information also leads to the notion of postponement (as in Sect. 4.2), where decision making is delayed until uncertain aspects of the future have passed. The illusion of control of uncertainty seems to have become a feature in every day life. During the winter 2003, local administrations along the Seine river (in Paris) were asked (in all seriousness) to begin planning for the impact of a flood expected to come approximately every 100 years because the last one had taken place in 1906! This is a perfect illustration of the misuse of statistics and of common misunderstanding of small probabilities (Kahneman and Tversky 1972).

1.5.2 Representation, Satisfaction Levels and Anchorage

We have discussed how human decision makers' attitude to risk seems to reverse around a point which we arbitrarily represented as zero in Fig. 1.2. It seems that each of us has a "neutral" point corresponding to our level of aspiration and that all our preferences are measured in reference to this point: aspiring to anything above it and rejecting anything below it. This idea is not new (Lewin et al. 1944; Siegel 1957) and was exploited by Tversky and Kahneman (1974). There is also ample empirical justification for it as it is clear that to lose or gain one euro is not the same for one of the Rockefellers or for a person sleeping rough.

The concept of level of aspiration is semantically close to that of level of reference which leads to the concept of anchoring. The point of anchoring is the point in relation to which the emotions and the experience of decision makers allow them to form an opinion and evaluate their choices. For example, a happy summer holiday in a Greek island will be used as point of reference to choose any future holiday. This phenomenon of anchoring has been noted to have several interesting dimensions: cognitive and mnesic dimensions, representational dimension and finally narrative dimension.

At the cognitive and mnesic level, certain events are ingrained in the memory and will affect future choices in situations where the decision making is emotionally reminded of them. This is very reminiscent of the "frame effect". A subject who had an unpleasant experience, even resulting from a good decision will hesitate when faced with the same decision. Individuals can be manipulated using their level of reference exactly as with the "frame effect". It even seems to be more effective than manipulating people's perception of context (Kühberger 1998).

Another less well-known effect, which is nonetheless well illustrated empirically and commonly used in AI, is the proximity effect whereby recent events (or freshly memorized) have a greater weight than older events. These recent events will greatly influence choices in looking for solutions to current problems. Anderson (1983) and Newell (1990) modelled this effect in an attempt to make their systems more credible. It has also been observed that the human mind is able to invent false correlations on the basis of completely independent events (Chapman and Chapman 1969). Thus, a simple experiment where certain figures are impressed on subjects before asking them the number of nations in the UN, will reveal that their cognition is influenced by the figures given to them, even though these bear no relationship whatever with the UN (Piattelli-Palmarini 1995). This phenomenon has also been interpreted as an anchoring effect (Tversky and Kahneman 1974). The proximity effect is a significant cognitive bias and it is particularly strong in the estimation of very small probabilities. Faced with a probability around $10^{-2}/10^{-4}$, a decision maker will be very likely to be overly influenced by recent events or salient events that are absolutely not related.

The second component of the anchoring effect is representational. It results in events with which one can easily relate being assigned a greater probability than those which are difficult to assimilate (Tversky and Kahneman 1982b; Slovic et al. 1982, 1988; March 1994, pp. 82–83). This is the representativeness effect. Morel (2002) has reported the case of an aircraft pilot who was so extremely anchored in considering the implication of his landing gear not deploying (which is not an uncommon situation) that he forgot the risk of running out of fuel until this eventually brought the plane down. This representativeness effect is very strong in terms of diagnosis as subjects' assessment of the current state of the world is unavoidably very dependent on their representations of it, which has lead to many reported accidents as in the case of the Three Mile Island near disaster so masterfully recalled in Perrow (1984). Other cases have been reported by Boy (1991) and Morel (2002).

The third component of the anchoring effect – undoubtedly the least known, is the narrative aspect. It is useful here to go back to our section on scenarios. To some extent, a scenario is a story. We have seen how, making a decision consists in inhibiting all possible scenarios except one, which will dominate and that this domination is established before the action (constructing a rationale for action see Pomerol and Adam 2003) or after the action (rationalization a posteriori). In any case, there is always a rationalization process, predominantly connected to the contextual elements of the decision (Brézillon et al. 2002). The more credible a story is, the more likely it is that the decision will be adopted. It is generally believed that the narrative mode is a fundamental mode of cognition going back thousands of years (Bruner 1986, 1990; Borland and Tenkasi 1995). It has therefore been written that, in order to "sell" a decision to organisational actors, one has to tell a story that everyone believes in (Weinberger 2001). The narrative side of decision making is a very distant relative of rationality but it brings us closer to language with which, as we said at the very beginning of this chapter rationality has many common features. As Vico (1744) indicated, before any theory of reasoning, mythologies and lyric poetry played a similar role, in the form of stories, as the first modes of structuring the world and accumulating knowledge. In the scientific era at any rate, Tversky and Kahneman (1982a) have shown how the easier to remember the stories the more likely the decision is to be successful (see also Kahneman and Lovallo 1993; Boland 1979).

This discussion begins to show the linkage between human decision modelling and artificial intelligence, in that it shows the importance of being able to represent cases of decision making (Anderson 1983; Newell 1990; Simon 1995). These three references illustrate that this topic has been of primary interest to the pioneers of artificial intelligence and the question remains open, on the one hand, whether one must approach "human reasoning" to the detriment of rationality and, on the other hand, what role the cognitive biases we have described here have played in the success of our species. Indeed, it is impossible to evaluate whether these biases and heuristic idiosyncrasies have conferred advantages or impediments to human beings in their fight for survival. Answering this question requires a multicriterion evaluation. Undoubtedly, decisions by heuristics have the advantage of speed and robustness, even if they do not have theoretical qualities (see Gigerenzer and Selten, Sect. 1.2.3). Speed is obviously an important factor for the survival of an individual as illustrated in very practical terms by Klein's "recognition-primed decision making" (see Sect. 1.3.3).

Other cognitive biases may no be so useful, even though cognitive psychology tends to view them in a positive light. At the end of the day, one must also realise that the rationality of the species as a whole is not necessarily the same as that of a given individual (the notion of survival of the fittest and the elimination of lesser males by the dominant males comes to mind here). Thus, it may be more difficult to justify the frame effect and the anchoring effects which allow the manipulation of other individuals in strict evolutionary terms. On the other hand, risk aversion for gains is certainly a useful behavioural aspect (prudence is a sound principle), but how can risk taking for the losses be useful? It may be that this cognitive anomaly is the strongest driver of cultural and technical change insofar as risk takers who throw caution to the wind are needed for innovation and great discoveries. Finally, it is possibly wise for a species to be able to neglect small probabilities and be able to find intellectual certainty in areas where there is none.

The most ambivalent cognitive bias is that of "risk control", because it leads to reckless behaviour in human activities (stock exchange, driving a car, etc...). But the other side of the coin is that, without this blindness to risk, there would probably have been no landing on the moon in 1969 (in an era where computers looked like fridges and had less computational power than the calculators today's school children use for their additions).

1.6 Conclusion

This chapter provides a historical tour of the link between decision theory and human decision making. The first observation we wish to make is that, contrary to the contentions of certain researchers in biology, psychology and sociology, these linkages run deep and are inherently useful.

These linkages run deep because for instance, if one ignored the principles put forwards by Savage's work, their limits, and their critical examination in light of today's understanding of qualitative decision making, it would simply be impossible to conceptualise the role of rationality in human decisions involving very uncertain situations. It is Savage's model that makes it possible to identify chance and bad luck in rational decisions. One could debate, as we did in this chapter, whether the assumptions of the model are realistic or rational, but one cannot deny that, apart from the game theories put forward in the 18th century, Savage's framework is the only quantitative or qualitative framework that provides the theoretical basis for distinguishing between a bad but lucky decision maker and a good but unlucky decision maker. Furthermore, it is the extension of this model by the psychologists Tversky and Kahneman that made it possible to understand the reversal of preferences in human decision makers facing uncertainty when they deal with losses or when they deal with gains.

The biological side of decision making is also a very interesting viewpoint from which to reflect on the specificities of human decision making, and therefore on the targets set for artificial intelligence. From an evolutionary point of view, even though it is clearly legitimate to consider language and decision making as specifically human activities, it is undeniable that there is a continuum between the neurons of the cockroach which make it "decide" to flee or to "play" dead, and our own neurons. Several millions of years of unfinished evolution explains the great complexity of the circuitry involved in decision making – as far as we can see, a seamless combination of different areas of the brain. Amongst the key areas, are the most primitive part of the brain and the prefrontal cortex, the most recently developed of all areas (see Berthoz 2003). Thus, human decision making is a team effort coupling the ancestral part of the brain, that most closely related to the body and the emotions, and the "reasoning" part with the frontal cortex acting as an integrating agent centralising all the information involved in the decision making process. The "reasoning" part gives us the capacity to project in the future, and by complex and, as yet, badly identified processes, to allow an action to eliminate all others within what amounts to a multi-criterion decision framework. Crucially, this domination sometimes requires a form of validation a posteriori, post-decision, because emotions and intuition can combine to allow for a quasi-instantaneous diagnosis of a situation or more complex pattern recognition guided by experience, followed by an immediate decision, which means that reason can only play catch up. Case-based reasoning and decision making and recognition primed decision making have allowed researchers to model this specificity of human decision making to a degree. In the end, it is elegant to conclude that the notion of expected utility, even though it applies in a strict sense in reality only in situations of risk, and then depends on the soundness of measured probabilities, is still the best way to represent rational decision making, in the same way that supply and demand in pure and perfect competition are the best way to represent the dynamics of markets, even though, in reality, no market behaves in the way these theories stipulate in the strict sense.

The limits of the notion of expected utility were pointed out with great clarity when Simon introduced his "counter-model" of bounded rationality. Bounded rationality offers, if not a complete model in the traditional sense of the term, a framework for understanding human reasoning, supporting the conception of and experimentation with many reasoning and problem solving systems, on the basis of existing models of decision making (Newell and Simon 1972; Anderson 1983, 1995; Newell 1990 inter alia). Simon's framework also provided the language and terminology to discuss heuristic searches and "what if" analysis. The simplicity of this framework and its explanatory power are certainly enough to explain its popularity and longevity.

Without revisiting our earlier observations on cognitive biases, we would urge that they be systematically taken into account in a rigorous way for any decision involving high stakes. And for anyone inclined to dismiss the normative models of decision making presented for instance by Simon (1977) and masterfully summarised by Mintzberg et al. (1976), it is useful to recall that without normative model, it is impossible to identify these biases. Consequently even when Savage is no great help, because probabilities are unknown and the utility function is not clearly defined, the theory can still act as a fortress against the ill effects of mis-representation of events, the illusion of risk control or the weakness of our minds when it comes to appreciating small probabilities.

Finally, we could not conclude this chapter without a warning for all the every-day decision makers and AI researchers: when all "i"s are dotted and all "t"s are crossed, and all possibility of bias have been pushed aside, one must still remember that a "decision is good only if it sounds good and tells a convincing story" (Sfez 1980, translated rather freely by the authors). This quote quite appropriately reminds us that the narrative and social dimensions of human decision making are the binds that tie decision and language, and make *Homo sapiens sapiens* a rather unique creature.

References

Adam F., Brézillon P., Humphreys P., Pomerol J.-Ch. (Eds.), 2002, *Decision Making and Decision Support in the Internet Age*, Proceedings of the IFIP W.G. 8.3 Conference, Oak Free Press, Cork, Ireland.

Adam F., Brézillon P., Courbon J.-Cl. (Eds.), 2003, Decision support systems in the internet age, *Journal of Decision Systems* 12(2).

Adolphs R., Tranel D., Bechara A., Damasio H., Damasio A.R., 1996, Neuropsychologigal approaches to reasoning and decision making, in: *Neurobiology of Decision-Making*, Damasio A.R., Damasio H., Christen Y. (Eds.), Springer, Berlin Heidelberg New York, pp. 157–179.

Allais M., 1953, Le comportement de l'homme rationnel devant le risque : critique des postulats et axiomes de l'école américaine, *Econometrica* 21, 503–546.

Anderson J.R., 1983, *The Architecture of Cognition*, Harvard University Press, Cambridge, MA.

Anderson J.R., 1995, *Cognitive Psychology and its Implications*, Freeman, New York.

Barki H., Rivard S., Talbot J., 1994, Risk management by information systems project leaders, *Journal of Management Information Systems* 10, 254–265.

Bell E.D., Raiffa H., Tversky A. (Eds.), 1988, *Decision Making*, Cambridge University Press, Cambridge, USA.

Berkeley D., Humphreys, P., 1982, Structuring decision problems and the 'bias heuristic'. *Acta Psychologia*, 50, 201–252.

Berthoz A., 1996, Neural basis of decision in perception and in the control of movement, in: *Neurobiology of Decision-Making*, Damasio A.R., Damasio H., Christen Y. (Eds.), Springer, Berlin Heidelberg New York, pp. 83–100.

Berthoz A., 2003, *La Décision*, Odile Jacob, Paris.

Boland Jr. R.J., 1979, Control, causality and information system requirements, *Accounting, Organizations and Society* 4, 259–272.

Boland Jr. R.J., Tenkasi R.V., 1995, Perspective making and perspective taking in communities of knowing, *Organization Science* 6, 350–372.

Bonczek R.H., Holsapple C.W., Whinston A.B., 1981, *Foundations of Decision Support Systems*, Academic Press, New York.

Bouchon-Meunier B., Marsala C. (Eds.), 2003, *Logique floue, principes, aide à la décision*, Vol. 1 and *Traitement des données complexes et commande en logique floue*, Vol. 2 *du traité Information-Commande-Communication*, Hermès-Lavoisier, Paris.

Bouchon-Meunier B., Nguyen H.T., 1996, *Les incertitudes dans les systèmes intelligents*, Que Sais-Je? no. 3110, PUF, Paris.

Boy G., 1991, *Intelligent Assistant Systems*, Academic Press, NY.

Brézillon P., Pasquier L., Pomerol J-Ch., 2002, Reasoning with contextual graphs, *European Journal of Operational Research* 136, 290–298.

Bruner J.S., 1986, *Actual Minds, Possible Works*, Harvard University Press, Cambridge, MA.

Bruner J.S., 1990, *Acts of Meaning*, Harvard University Press, Cambridge MA.

Burstein F. (Ed.), 2001, Decision support in the new millennium, *Journal of Decision Systems* 10(3).

Calvin W.-H., 1991, *The Ascent of Mind: Ice Age, Climate and the Evolution of Intelligence*, Bantam Books.

Calvin W.-H., 1994, La naissance de l'intelligence, *Pour la Science* 206, 110–117.

Chapman L.J., Chapman J.P., 1969, Illusory correlation as an obstacle to the use of valid psychodiagnostic signs, *Journal of Abnormal Psychology* 74, 271–280.

Cohen M., Jaffray J.-Y., Said T., 1987, Experimental comparison of individual behavior under risk and uncertainty for gains and losses, *Organizational Behavior and Human Decision Processes* 39, 1–22.

Cohen M., Jaffray J.-Y., 1988, Is Savage's independance axiom a universal rationality principle?, *Behavioral Science* 33, 38–47.

Cohen D.M., March J.G., Olsen J.P., 1972, A garbage can model of organizational choice, *Administrative Science Quarterly* 17, 1–25.

Damasio A.R., 1994, *Descartes' Error*, Putnam's Sons, New York.

Damasio A.R., Damasio H., Christen Y. (Eds.), 1996, *Neurobiology of Decision-Making*, Springer, Berlin Heidelberg New York.

Dempster, A.-P., 1967, Upper and lower probabilities induced by a multivalued mapping, *Annals of Mathematical Statistics* 38, 325–339.

Dubois D., Prade H., 1985, *Théorie des Possibilités*, Masson, Paris. *Possibility Theory, An Approach to Computerized Processing of Uncertainty* (traduction anglaise), Plenum Press, NY, USA.

Dubois D., Prade H., 1995, Possibility theory as a basis for qualitative decision theory, *Proceedings IJCAI-95*, Montréal, 1924–1930.

Dubois D., Prade H., Smets Ph., 1996, Representing partial ignorance, *IEEE Transactions on Systems, Man, and Cybernetics* 26, 361–377.

Dubois D., Prade H., Sabbadin R., 2001, Decision–theoretic foundations of qualitative possibility theory, *European Journal of Operational Research* 128, 459–478.

Dubois D., Fargier H., Prade H., Perny P., 2002, Qualitative decision theory: from Savage's Axiomes to Nonmonotonic Reasoning, *Journal of the ACM* 49(4), 455–495.

Dubois D., Fargier H., Perny P., 2003, Qualitative decision theory with preference relations and comparative uncertainty: an axiomatic approach, *Artificial Intelligence* 148, 219–260.

Eisenhardt K.M., 1990, Speed and strategic choice: how managers accelerate decision making, *California Management Review* 31, 39–54.

Essid S., 1997, Choice under risk with certainty and potential effects: a general axiomatic model, *Mathematical Social Science* 34, 223–247.

Festinger L., 1957, A theory of cognitive dissonance, Stanford University Press, Stanford.

de Finetti B., 1937, La prévision: ses lois logiques, ses sources subjectives, *Annales de l'Institut Henri Poincaré* 7, 1–68.

Fuster J.M., 1996, Frontal lobe and the cognitive foundation of behavioural action, in: *Neurobiology of Decision-Making*, Damasio A.R., Damasio H., Christen Y. (Eds.), Springer, Berlin Heidelberg New York, pp. 115–123.

Gigerenzer G., 2002, The adaptive toolbox, in: *Bounded Rationality: The Adaptive Toolbox*, Gigerenzer G., Selten R. (Eds.), MIT Press, Cambridge, MA, pp. 37–50.

Gigerenzer G., Selten R., 2002, Rethinking rationality, in: *Bounded Rationality: The Adaptive Toolbox*, Gigerenzer G., Selten R. (Eds.), MIT Press, Cambridge, MA, pp. 1–12.

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 Cybernetics* 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.

Grenier B., 1990, *Evaluation de la décision médicale*, Masson, 3ième édition, 1999, Paris.

Howard R.A., 1988, Decision analysis: practice and promise, *Management Science* 34, 679–695.

Humphrey P., Bekerley D., 1985, Handling uncertainty: levels of analysis of decision problems, in: G. Wright (Ed.), *Behavioural Decision Making*, Plenum Press, London.

Humphreys and Brézillon P. (Eds.), 2001, *Decision systems in action*, Journal of Decision Systems 10(2).

Jaffray J.-Y., 1988, Choice under risk and security factor. An axiomatic model, *Theory and Decision* 24, 169–200.

Janis I.L., Mann L., 1977, *Decision Making: a Psychological Analysis of Conflict, Choice and Commitment*, Free Press, New York.

Judge A., 1997, The *Art of Non-Decision Making and the manipulation of categories*, http://www.uia.org/uiadocs/nondec.htm 24/10/2001.

Kahneman D., Lovallo D., 1993, Timid choices and bold forecast: a cognitive perspective on risk taking, *Management Science* 39, 17–31.

Kahneman D., Tversky A., 1972, Subjective probability: a judgment of representativeness, *Cognitive Psychology* 3, 430–454.

Kahneman D., Tversky A., 1979, Prospect theory, an analysis of decision under risk, *Econometrica* 47, 263–291.

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, UK.

Keen PGW, 1977, The evolving concept of optimality, in: *Multiple Criteria Decision Making*, Starr M.K., Zeleny M., (Eds.), *TIMS Study in Management Science* 6, North Holland, pp. 31–57.

Keen P.G.W., Scott-Morton M.S., 1978, *Decision Support Systems*, Addison-Wesley, Reading, MA.

Keeney R.-L., 1988, Value-driven expert systems for decision support, *Decision Support Systems*, 4, 405–412.

Keeney R.L., 1992, *Value-Focused Thinking*, Harvard University Press, Cambridge, MA.

Klein G.A., 1993, A recognition-primed decision (RPD) model of rapid decision making, in *Decision Making in Action, Models and Methods*, Klein G.A., Orasanu J., Calderwood R., Zsambok C.E., (Eds.), Ablex, Nordwood, NJ, pp. 138–147.

Klein G.A., 2002, The fiction of optimization, in: *Bounded Rationality: The Adaptive Toolbox*, Gigerenzer G., Selten R. (Eds.), MIT Press, Cambridge, MA, pp. 103–121.

Kleinmuntz D.N., 1985, Cognitive heuristics and feedback in a dynamic decision environment, *Management Science* 31, 680–702.

Kleinmuntz D.N., 1993, Information processing and misperceptions of the implications of feedback in dynamic decision making, *System Dynamics Review* 9, 223–237.

Knight, F.H., 1921, *Risk Uncertainty and Profit*, 1964 [original 1921] Augustus M. Kelley, New Impression of 1921 Edition.

Kolodner J., 1993, *Case-Based Reasoning*, Morgan Kaufmann, San-Francisco.

Kottemann J.E., Davis D.R., 1991, Decisional conflict and user acceptance of multicriteria decision-making aids, *Decision Sciences* 22, 918–926.

Kühberger A., 1998, The influence of framing on risky decisions: a meta-analysis, *Organizational Behavior and Human Decision Processes* 75, 23–55.

Lehmann D., 1996, Generalized qualitative probability: savage revisited, in the *Proceedings of the Conference on Uncertainty in AI*, pp. 381–388.

Leland J.W., 1994, Generalized similarity judgments: an alternative explanation for choices anomalies, *Journal of Risk and Uncertainty* 9, 151–172.

Lévine P., Pomerol J.-Ch., 1986, PRIAM an interactive method for choosing among multiple attribute alternatives, *European Journal of Operational Research* 25, 272–280.

Lévine P., Pomerol J.-Ch., 1989, *Systèmes Interactifs d'aide à la décision et systèmes experts*, Hermès, Paris.

Lévine P., Pomerol J.-Ch., 1995, The role of the decision maker in DSSs and representation levels, in: J.F. Nunamaker Jr., R.H. Sprague (Eds.), *Proceedings. Of 26th Hawaï International Conference on System Sciences*, Vol. 3, IEEE, pp. 42–51.

Lewin K., Dembo T., Festinger L., and Sears P., 1944, Level of aspiration, in: *Personality and the Behavior Disorders*, J.M. Hunts (Ed.), Ronald Press, New York, pp. 333–378.

March J.G., 1994, *A Primer on Decision Making*, The Free Press, New York.

March J.G., Olsen J.P. (Eds.), 1976, *Ambiguity and Choice in Organizations*, Universitetsforlaget, Bergen.

March J.G., Shapira Z., 1987, Managerial perspectives on risk and risk taking, *Management Science* 33, 1404–1418.

March J.G., Simon H.A., 1993, *Organizations* (seconde édition; première édition 1958), Basil Backwell, Oxford, UK.

McKenna F.P., 1993, It won't happen to me: unrealistic optimism or illusion of control, *British Journal of Psychology* 84, 39–50.

Mintzberg, H., Raisinghani, D., Theoret, A., 1976, The structure of "unstructured" decision processes, *Administrative Science Quarterly* 21, 246–275.

Monnet J.-M., Lagrange J.-Ph., Pomerol J.-Ch., Teulier R., 1998, A formal framework for decision-making in case-based reasoning. *Proceedings IPMU*, Paris, Vol. 2, pp. 1274–1281.

Montgomery H., 1983, Decision rules and the search for a dominance structure : towards a process model of decision making, in: *Analysing and Aiding Decision Processes*, Humphreys, P.C., Svenson O., Vari A., (Eds.), North-Holland, Amsterdam, pp. 343–369.

Montgomery H., 1987, Image theory and dominance search theory: How is decision making actually done? *Acta Psychologica* 66, 221–224.

Mora M., Forgionne G., Gupta J. (Eds.), 2002, *Decision Making Support Systems*, Idea Group Publishing, Hershey, PA.

Morel C., 2002, *Les décisions absurdes*, Galimard, Paris.

Nau R.F., 2001, De Finetti was right: probability does not exist, *Theory and Decision* 51, 89–124.

Newell A., 1990, *Unified Theories of Cognition*, Harvard University Press, Cambridge, MA.

Newell A., Simon H.A., 1972, *Human Problem Solving*, Prentice-Hall, Englewood Cliffs, NJ.

Oliver R.M., Smith J.Q., (Eds.), 1990, *Influence Diagrams, Belief Nets and Decision Analysis*, Wiley, Chichester.

Perrow C., 1984, *Normal Accidents*, Basic Books, NY. Réédition augmentée Princeton University Press, 1999, Princeton, NJ.

Piattelli-Palmarini M., 1995, *La réforme du jugement ou comment ne plus se tromper*, Odile Jacob, Paris.

Pomerol J-Ch., 1997a, Artificial intelligence and human decision making, *European Journal of Operational Research* 99, 3–25.

Pomerol J-Ch., 1997b, Cognition and decision: about some recent results in neurobiology, in: *ISDSS97 Proceedings*, Lausanne, pp. 115–125.

Pomerol J-Ch., 2001, Scenario development and practical decision making under uncertainty, *Decision Support Systems* 31, 197–204.

Pomerol J-Ch., 2002, L'apport de Herbert Simon dans le management et la décision, *Revue d'Intelligence Artificielle* 16, 221–249.

Pomerol J-Ch., 2003, Decision Making Biases and Context, *Journal of Decision Systems* 12, no. 3/4, 235–252.

Pomerol J-Ch., Adam F., 2002, From human decision making to DMSS architecture, in: *Decision Making Support Systems*, Mora M., Forgionne G., Gupta J., (Eds.), pp. 40–70.

Pomerol J-Ch., Adam F., 2003, Multi-level and Multi-model DSSs, in: *DSS in the Uncertainty of the Internet Age*, T. Bui, H. Sroka, S. Stanek, J. Goluchowski (Eds.), Karol Adamiecki University of Economics Press, Katowice, Poland, pp. 333–346.

Pomerol J-Ch., Barba-Romero, S., 1993, *Choix multicritère dans l'entreprise*, Hermès, Paris.

Pomerol J-Ch., Roy B., Rosenthal-Sabroux S.C., Saad A., 1995, An intelligent DSS for the multicriteria evaluation of railway timetables, *Foundations of Computer and Decision Science* 20/3, 219–238.

Pomerol J-Ch., Brézillon P., Pasquier L, 2002, Operational representation for practical decision making, *Journal of Management Information Systems* 18(4), 101–116.

Poundstone W., 1990, *Les labyrinthes de la raison*, Belfond, Paris.

Raiffa H.A., 1968, *Decision Analysis*, McGraw-Hill, New-York.

Riesbeck C.K., Schank R.C., 1989, *Inside case-based Reasoning*, Laurence Erlbaum, Hillsdale, NJ

Rosenhead J., 2001, Robustness analysis: keeping your options open, in: Rosenhead J., Mingers J., (Eds.), *Rational Analysis for a Problematic World Revisited: problem Structuring Methods for Complexity Uncertainty and Conflict*, Wiley, Chichester, pp. 181–207.

Roy B., 1998, A missing link in operational research decision aiding: robustness analysis, *Foundations of Computer and Decision Science* 23, 141–160.

Roy B., 2000, Réflexions sur le thème, quête de l'optimum et aide à la décision, in: *Decision, Prospective et Auto-Organisation, Mélanges en l'honneur de Jacques Lesourne*, Thépot J., Godet M., Roubelat F., Saad A.e., (Eds.), Dunod, Paris, pp. 61–83.

Roy B., 2002, Robustesse de quoi et vis-à-vis de quoi, mais aussi robustesse pourquoi en aide à la décision? *Proceedings of the 56th Meeting of the European Working Group on Multiple Criteria Decision Aiding*, Coimbra, October 2002.

Rubinstein A., 1988, Similarity and decision-making under risk (is there a utility theory resolution to the Allais paradox?), *Journal of Economic Theory* 46, 145–153.

Savage L.J., 1954, *The Foundations of Statistics*, Second Edition, Dover Publications, 1972, New York.

Schelling T.C., 1962, Preface in: R. Wohlstetter: *Pearl Harbour –Warning and Decision*, Stanford University Press, Stanford. Traduction française: Pearl Harbour n'était pas possible, Stock, 1964.

Schoenbaum G., Chiba A., Gallagher M., 1998, Orbito-frontal cortex and basolateal amygdala encode expected outcomes during learning, *Nature Neurosciences* 1, 155–159.

Selten R., 2002, What is bounded rationality, in: *Bounded Rationality: The Adaptive Toolbox*, Gigerenzer G., Selten R., (Eds.), MIT Press, Cambridge, MA, pp. 13–36.

Sfez L., 1980, *Je reviendrai des terres nouvelles*, Hachette Littérature, Paris.

Shafer G., 1976, *A Mathematical Theory of Evidence*, Princeton University Press, Princeton, NJ.

Shafer G., 1996, *The Art of Causal Conjecture*, MIT Press, Cambridge, MA.

Shafir E.B., Osherson D.N., Smith E.E., 1993, The advantage model: a comparative theory of evaluation and choice under risk, *Organisational Behavior and Human Decision Processes* 55, 325–378.

Shenoy P., 1994, A comparison of graphical techniques for decision analysis, *European Journal of Operational Research* 78, 1–21.

Siegel S., 1957, Level of aspiration and decision making, *Psychological Review* 64, 253–262.

Simon H.A., 1955, A behavioural model of rational choice, *Quarterly Journal of Economics* 69, 99–118.

Simon H.A., 1972, Theories of bounded rationality, in: *Decision and Organization*, McGuire C.B., Radner R., (Eds.), North Holland, Amsterdam, pp. 161–176.

Simon H.A., 1976, From substantive to procedural rationality, in: *Methods and Appraisal in Economics*, Latsis S.J. (Ed.), Cambridge University Press, Cambridge, NY, pp. 129–148.

Simon H.A., 1977, *The New Science of Management Decision* (3ième édition révisée; première édition 1960), Prentice-Hall, Englewood Cliffs, NJ.

Simon H.A., 1983, *Reason in Human Affairs*, Basil Blackwell, Oxford, UK.

Simon H.A., 1984, Commentaires dans le cadre du Colloque de la Grande Motte, in *Sciences de l'Intelligence, Sciences de l'Artificiel*, Demailly A., Le Moigne J.L., (Eds.), Presses Universitaires de Lyon, 1986, pp. 577–619.

Simon H.A., 1991, *Models of My Life*, Basic Books, London.

Simon H.A., 1995, Explaining the ineffable: AI on the Topics of Intuition, Insight and Inspiration, *Proceedings of IJCAI*-95, pp. 939–948.

Simon H.A., 1997, *Administrative Behavior* (4th edition, first edition 1947), The Free Press, NY.

Slovic P., Fischhoff B., Lichtenstein S., 1982, Facts versus fears: understanding perceived risk, in: *Judgment Under Uncertainty: Heuristics and Biases*, Kahneman D., Slovic P., Tversky A. (Eds.), Cambridge University Press, Cambridge, UK, pp. 463–489.

Slovic P., Fischhoff B., Lichtenstein S., 1988, Response mode, framing and information-processing effects in risk assessment, in: *Decision Making*, Bell D.E., Raiffa H., Tversky A. (Eds.), Cambridge University Press, Cambridge, MA, pp. 152–166.

Slovic P., Finucane M., Peters E., MacGregor D.G., 2002, The affect heuristic, in: *Heuristics and Biases: the psychology of Intuitive Judgment*, Gilovich T., Griffin D., Kahneman D. (Eds.), Cambridge University Press, Cambridge, MA, pp. 397–420.

Smets Ph., 1990, Constructing the pignistic probability function in a context of uncertainty, in: *Uncertainty in Artificial Intelligence*, Heurion M., Schachter D., Kanal N., Leurmer J., (Eds.), Vol. 5, North-Holland, Amsterdam, pp. 29–40.

Sprague R.H., Carlsson E.D., 1982, *Building Effective Decision Support Systems*, Prentice-Hall, NJ.

Sterman J.-D., 1989, Modelling managerial behaviour: Misperceptions of feedback in a dynamics decision-making experiment, *Management Science* 35, 321–339.

Tversky A., Kahneman D., 1974, Judgment under uncertainty: Heuristics and Biases, *Science* 185, 1124–1131.

Tversky A., Kahneman D., 1982a, Subjective probability: a judgment of representativeness, in: *Judgment Under Uncertainty: Heuristics and Biases*, Kahneman D., Slovic P., Tversky A. (Eds.), Cambridge University Press, Cambridge, UK, pp. 32–47.

Tversky A., Kahneman D., 1982b, Availability: a heuristic for judging frequency and probability, in: *Judgment Under Uncertainty: Heuristics and Biases*, Kahneman D., Slovic P., Tversky A. (Eds.), Cambridge University Press, Cambridge, UK, pp. 163–178.

Tversky A., Kahneman D., 1983, Extensional versus intuitive reasoning: the conjunction fallacy in probability judgment, *Psychological Review* 90, 293–315.

Tversky A., Kahneman D., 1988, Rational choice and the framing of decisions, in: *Decision Making*, Bell D.E., Raiffa H., Tversky A. (Eds.), Cambridge University Press, Cambridge, MA, pp. 167–192.

Tversky A., Simonson I., 1993, Context-dependent preferences, *Management Science* 39, 1179–1189.

Tversky A., Wakker P., 1995, Risk attitudes and decision weights, *Econometrica* 63, 1255–1280.

Vico G., 1744, *Principes d'une science nouvelle relative à la nature commune des nations*, traduction 1986, Nagel, Paris.

Vincke Ph., 1999, Robust solutions and methods in decision aid, *Journal of Multicriteria Decision Analysis* 8, 181–187.

Weinberger D., 2001, Garbage in great staff out, *Harvard Business Review* 79 *no.* 8, 30–32.

von Winterfeldt D., Edwards W., 1986, *Decision Analysis and Behavioral Research*, Cambridge University Press, Cambridge, MA.

Woolsey R. 1991, The fifth column: La Méthode de Charles Maurice de Talleyrand or maximized acceptance with optimized agendas, *Interfaces* 21, 103–105.

Zickar M.J., Highhouse S., 1998, Looking closer on the effects of the framing on risky choice: an item response theory analysis, *Organizational Behavior and Human Decision Processes* 75, 75–91.

Zionts S., 1992, The state of multiple criteria decision making: past, present and future, in: *Multiple Criteria Decision Making*, Goicoechea A., Duckstein L., Zionts S., (Eds.), Springer, Berlin Heidelberg New York, pp. 33–43.

2

Cognitive Elements of Human Decision Making

Jens Pohl

Collaborative Agent Design Research Center, California Polytechnic State University, San Luis Obispo, California, USA

Summary. This chapter presents some understandings of the human problem-solving activity that a group of researchers in the Collaborative Agent Design Research Center at the California Polytechnic State University, San Luis Obispo, California has gained over the past two decades. Based on the premise that the human decision-maker should be an integral component of any computer-based decision-support system, it follows that the elements that appear to be important to the user should be incorporated in the design of these systems. The complexity of the human cognitive system is evidenced by the large body of literature that describes problem-solving behavior and the relatively fewer writings that attempt to provide comprehensive explanations of this behavior. The contributions of this chapter are confined to the identification of important elements of the problem-solving activity and exploration of how these elements might influence the design of a decision-support system.

2.1 Introduction

One could argue that among all attributes it is the intellectual capabilities that have allowed human beings to gain superiority over all animal species on planet Earth. These intellectual capabilities have allowed us humans to adapt to our environment, protect ourselves from predators, and ensure the availability of an adequate supply of food and water under all but the most extreme circumstances. Furthermore, these intellectual capabilities have evolved from very primitive beginnings into much more sophisticated and specialized skills (e.g., observation, planning, coordination, and problem solving), over the relatively short period of a few thousand years.

J. Pohl: *Cognitive Elements of Human Decision Making*, Studies in Computational Intelligence (SCI) **97**, 41–76 (2008)
www.springerlink.com

In this chapter[1] the author will explore the strengths and limitations of human beings within the context of an evolving information society in which the ability to analyze problem situations, formulate and evaluate solution alternatives, and make accurate and timely decisions, are highly valued attributes. It would appear appropriate that the designers and developers of intelligent software systems should not only seek to advance the state-of-the-art of artificial intelligence (AI), but also consider how AI-based decision-support systems can best compliment human decision-making capabilities. In this respect it is also necessary to explore the problem solving techniques that we devised before the advent of computer technology, over many centuries, to suit our human cognitive capabilities. The implication is that computers may be a necessary component of human evolution by overcoming some of the limitations of our intellectual assets.

2.2 Will Technology and Biology Merge?

The principal enabling characteristics of the Information Society are revolutionary advances in computer, bio-electronic, and communication technologies. By utilizing these technological advances a single person is able to achieve today what entire organizations struggled to accomplish only three decades ago. However, at the same time, these new opportunities are placing unprecedented pressure on the individual to perform at a significantly higher level of expectation. How will the human intellectual capabilities that have served us so well in the past measure up in this new era of unprecedented opportunities and corresponding expectations? To what degree can AI-based software systems extend our intellectual capabilities and where should this assistance be best focused?

Kurzweil (1999) argues rather convincingly that technology and biology will merge over the next millennium to significantly accelerate human evolution. Recent developments in subcutaneous sensors and prosthetics (Finn, 1997), bio-engineered materials (Kelly, 1994), brain scanning (Kiernan, 1998; Hübener, 1997; Powledge, 1997), and unraveling of the human genome (DOE, 2000), appear to be only the beginning of bio-electronic advances that promise profound extensions to the quality, productivity and longevity of human life (Brooks, 2002). In Kurzweil's words (Brockman, 2002) "... We are entering a new era. I call it the Singularity. It's a merger between human intelligence and machine intelligence ..."

---

[1] Portions of the material included in this chapter have appeared previously in keynote addresses presented at two InterSymp Conferences (Pohl, 2002, 2006) and two Technical Reports of the Collaborative Agent Design Research Center at the California Polytechnic State University, San Luis Obispo, California (Pohl et al., 1994, 1997)

2.3 Some Human Problem Solving Characteristics

Human beings are inquisitive creatures by nature who seek explanations for all that they observe and experience in their living environment. While this quest for understanding is central to our success in adapting to a changing and at times unforgiving environment, it is also a major cause for our willingness to accept partial understandings and superficial explanations when the degree of complexity of the problem situation confounds our mental capabilities. In other words, a superficial or partial explanation is considered better than no explanation at all. As flawed as this approach may be, it has helped us to solve difficult problems in stages. By first oversimplifying a problem we are able to develop an initial solution that is later refined as a better understanding of the nature of the problem evolves. Unfortunately, this requires us to contend with another innately human characteristic, our inherent resistance to change and aversion to risk taking. Once we have found an apparently reasonable and workable explanation or solution we tend to lose interest in pursuing its intrinsic shortcomings and increasingly believe in its validity. Whether driven by complacency or lack of confidence, this state of affairs leads to many surprises. We are continuously discovering that what we believed to be true is only partly true or not true at all, because the problem is more complicated than we had previously assumed it to be.

The complexity of the problems faced by human society in areas such as management, economics, marketing, engineering design, and environmental preservation, is increasing for several reasons. First, computer-driven information systems have expanded these areas from a local to an increasingly global focus. Even small manufacturers are no longer confined to a regionally localized market for selling their products. The marketing decisions that they have to make must take into account a wide range of factors and a great deal of knowledge that is far removed from the local environment. Second, as the net-centricity of the problem system increases so do the relationships among the various factors. These relationships are difficult to deal with, because they require the decision-maker to consider many factors concurrently. Third, as the scope of problems increases decision-makers suffer simultaneously from two diametrically opposed but related conditions. They tend to be overwhelmed by the shear volume of data that they have to consider, and yet they lack information in many specific areas. To make matters worse, the information tends to change dynamically in largely unpredictable ways.

It is therefore not surprising that governments, corporations, businesses, down to the individual person, are increasingly looking to computer-based decision-support systems for assistance. This has placed a great deal of pressure on software developers to rapidly produce applications that will overcome the apparent failings of the human decision-maker. While the expectations have been very high, the delivery has been much more modest. The expectations were simply unrealistic. It was assumed that advances in technology would be simultaneously accompanied by an understanding of how these advances should be applied optimally to assist human endeavors. History suggests that such an a priori assumption is not justified. There are countless examples that would suggest the contrary. For example, the invention of new materials (e.g., plastics) has inevitably been followed by a period of misuse. Whether based on a misunderstanding or lack of knowledge of its intrinsic properties, the new material was typically initially applied in a manner that emulated the material(s) it replaced. In other words, it took some time for the users of the new material to break away from the existing paradigm. A similar situation currently exists in the area of computer-based decision-support systems.

2.4 Human Limitations and Weaknesses

Deeply embedded in the evolution of the human intellect is the rationalistic approach to problem solving. At face value this approach appears to be entirely sound. It suggests that problem solving should proceed in a logical sequence of clearly defined steps. One begins by defining the problem and then decomposes the defined problem into sub-problems. Decomposition appears to make a great deal of sense because the parts of a problem are intrinsically easier to solve than the whole problem. The reason for this is that the complexity of a problem is normally due to the nature and number of relationships among the elements of the problem and not due to the elements themselves. Decomposition allows us to temporarily neglect consideration of many of these relationships. However, this over-simplification of the problem is valid only as long as the problem remains in a decomposed state. As soon as we try to integrate the separate solutions of the parts into a solution of the whole the relationships that we so conveniently disregarded reappear and invalidate many if not most of our neatly packaged sub-solutions. We find to our consternation that the characteristics of a part of a problem situation considered in isolation are not necessarily similar (let alone the same) as the behavior of that part within the context of the whole problem.

Why have we human beings come to rely so heavily on this flawed approach to problem solving? The reasons are related primarily to the biological nature of our cognitive system. While the biological basis of human cognition is massively parallel (i.e., millions of neurons and billions of connections) our conscious reasoning capabilities are largely sequential. The fact is that our short term memory has a severely limited capacity of only a few chunks of data at any one time. Therefore, we can differentiate among only a small number of objects at any one point in time, even though we continuously move new data chunks from long term memory into short term memory. As a consequence we have great difficulty dealing with more than three or four relationships concurrently.

Secondary limitations and tensions that contribute to our human problem solving difficulties include our tendency to seek a degree of accuracy that is often unrealistic and usually unnecessary. Our aversion to risk and instinctive need to survive drives us to try to predict the future with great accuracy. In this respect, as mentioned previously, we place a great deal of reliance on mathematics even though mathematical models often fail due to oversimplification of the problem situation and incorrect boundary assumptions (Pohl, 1999).

We often seek to produce an optimum solution even though the problem conditions are continuously changing and, therefore, we have no benchmark that would allow us to judge whether a particular solution is in fact optimal. In other words, under dynamic conditions there is no static benchmark available. This creates related difficulties, because our ability to interpret and judge any situation is necessarily based on comparative analysis. Subject to the experiential basis of the human cognitive system we normally have no alternative but to measure new situations with existing metrics based on past experience. However, the further the new situation deviates from past experience the more misleading the available metrics are likely to be. As a result, since we have no effective metrics for assessing new situations, we typically require a considerable period of time to correctly evaluate such situations. Accordingly, it is not unreasonable to conclude that human judgments are more influenced by the past than the present.

More comprehensively stated, the essentially experience-based nature of human cognition forces us almost always (i.e., at least initially) to apply existing methods, notions, and concepts to new situations. Therefore, our most effective problem solving capabilities utilize prototype solutions based on past experience. While we have become quite skilled in adapting, modifying and combining such prototype solutions, we find it very difficult to create new prototypes. As a consequence we invariably apply existing solution methods to new problem situations and develop new methods only through painful trial and error. This also leads us to generally underestimate the complexity and impact of new situations.

2.5 Human Strengths

So far the discussion has centered on the apparently numerous limitations and weaknesses of human beings, particularly in respect to intellectual and emotional capabilities. Surely we human beings also have intellectual strengths. The answer is yes, of course, but with some qualifications. Certainly human learning capabilities, supported by a very large associative long-term memory, are vast. However, our rate of learning is rather slow and appears to lack efficiency. While some of this inefficiency is undoubtedly due to human communication inadequacies, the very process of progressively collecting experience by building onto existing associative knowledge structures would appear to be cumbersome and rather time consuming. It is not simply a matter of adding new knowledge elements or associating existing elements by inserting linkages, but instead patterns of neural activations (i.e., firings) have to be repeated many times before they are literally grooved into long-term memory. It is therefore not surprising that formal education takes up one quarter to one third of a human life span and involves a great deal of concentration, as well as assistance from other human beings who have acquired special teaching skills.

An important part of the human learning capability is the ability to conceptualize experiences into knowledge that we consider to be true in most cases. In this way we place emphasis on being able to deal with general conditions and consider the exceptions to the general rules to be much less important. This again exemplifies the human tendency to oversimplify a situation for the sake of being able to reach a quick solution to a problem or an explanation of an observed phenomenon. In fact, as we discover to our surprise time and again, the exceptions are often more important than the generalizations (Minsky, 1990).

It must also be noted that much of human learning is involuntary and therefore virtually effortless. This applies in particular to the acquisition of low-level, largely involuntary skills such as sensory pattern matching that allows us to automatically convert data to information. For example, when we enter a restaurant we immediately recognize the furniture in the room. In fact, our eyes see only image patterns. However, these are automatically interpreted as tables and chairs by our cognitive system which has by experience related these image patterns to the appropriate symbolic entities.

At a higher level, symbolic reasoning allows us to infer knowledge from information. When our reasoning capabilities are unable to cope in complex situations that include many relationships, conceptual pattern matching (i.e., intuition) allows us to assess situations without resorting to logical reasoning. However, again there is evidence that this process is greatly facilitated by experience. Klein (1998) found that highly experienced fire captains will resort to the situation analysis methods employed by novices when they are confronted with situations outside their sphere of experience.

While the creation of new knowledge is normally the province of individuals, once such an intellectual leap has been accomplished we collectively excel in the technological exploitation of this contribution. Typically, this exploitation proceeds incrementally and involves a large number of persons, coordinated in a self-organizing fashion but willing to collaborate to leverage the capabilities of individual contributors.

However, finally, perhaps one of our greatest human strengths is the discovery early on in our evolution of the usefulness of tools. Since then we have been successful in the development and application of more and more powerful tools. Today, we appear to be on the verge of merging computer-based tools with the biological fabric of our very existence.

2.6 The Rationalistic Tradition

To understand current trends in the evolution of progressively more sophisticated decision-support systems it is important to briefly review the foundations of problem solving methodology from an historical perspective. Epistemology is the study or theory of the origin, nature, methods and limits of knowledge. The dominant epistemology of Western Society has been technical rationalism (i.e., the systematic application of scientific principles to the definition and solution of problems).

The rationalistic approach to a problem situation is to proceed in well defined and largely sequential steps as shown in Fig. 2.1: define the problem; establish general rules that describe the relationships that exist in the problem system; apply the rules to develop a solution; test the validity of the solution; and, repeat all steps until an acceptable solution has been found. This simple view of problem solving suggested a model of sequential decision-making that has retained a dominant position to the present day. With the advent of computers it was readily embraced by first Wave software (Fig. 2.2) because of the ease with which it could be translated into decision-support systems utilizing the procedural computer languages that were available at the time.

The close correlation between the rationalistic approach and what is commonly referred to as the scientific method is readily apparent in the series of basic steps that are employed in scientific investigations: observe the phenomenon that requires explanation; formulate a possible explanation; develop a method capable of predicting or generating the observed phenomenon; interpret the results produced by the method; and, repeat all steps until an acceptable explanation of the observed phenomenon has been found. Scientific research typically attempts to establish situations in which observable actions (or reactions) are governed by a small number of variables that can be systematically manipulated. Every effort is made to keep the contrived situation

STEP 1:	Define problem as a system of identifiable objects that have known characteristics.
STEP 2:	Find general rules that define the relationships among the objects within the context of the problem system.
STEP 3:	Apply the rules to the problem situation and draw conclusions that lead to a solution.
STEP 4:	Test the solution against specific acceptance criteria and if unsatisfactory, return to any of the previous steps.

Fig. 2.1. Solution of simple problems

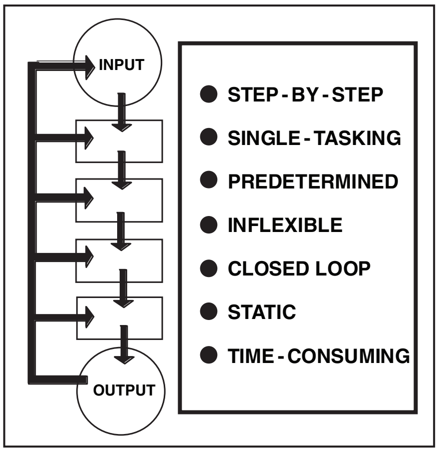


Fig. 2.2. Sequential decision-support

simple, clear and deterministic, so that the results of the simulation can be verified.

However, natural phenomena and real world problems are often very complex involving many related variables. Neither the relationships among the variables nor the variables themselves are normally sufficiently well understood to provide the basis for clear and comprehensive definitions. In other words, problem situations are often too complex to be amenable to an entirely logical and predefined solution approach. Under these circumstances the analytical strategy has been to decompose the whole into component parts, as follows:

- Decompose the problem system into sub-problems.
- Study each sub-problem in relative isolation, using the rationalistic approach (Fig. 2.1). If the relationships within the sub-problem domain cannot be clearly defined then decompose the sub-problem further.
- Combine the solutions of the sub-problems into a solution of the whole.

Underlying this problem-solving strategy is the implicit assumption that an understanding of parts leads to an understanding of the whole. Under certain conditions this assumption may be valid. However, in many complex problem situations the parts are tightly coupled so that the behavior of the whole depends on the interactions among the parts rather than the internal characteristics of the parts themselves (Bohm, 1983; Senge, 1993). An analogy can be drawn with the behavior of ants. Each ant has only primitive skills, such as the ability to interpret the scent of another ant and the instinctive drive to search for food, but little if any notion of the purpose or objectives of the ant colony as a whole. In other words, an understanding of the behavior of an individual ant does not necessarily lead to an understanding of the community behavior of the ant colony of which the ant is a part.

Decomposition is a natural extension of the scientific approach to problem solving and has become an integral and essential component of rationalistic methodologies. Nevertheless, it has serious limitations. First, the behavior of the whole usually depends more on the interactions of its parts and less on the intrinsic behavior of each part. Second, the whole is typically a part of a greater whole and to understand the former we have to also understand how it interacts with the greater whole. Third, the definition of what constitutes a part is subject to viewpoint and purpose, and not intrinsic in the nature of the whole. For example, from one perspective a coffee maker may be considered to comprise a bowl, a hotplate, and a percolator. From another perspective it consists of electrical and constructional components, and so on.

Rationalism and decomposition are certainly useful decision-making tools in complex problem situations. However, care must be taken in their application. At the outset it must be recognized that the reflective sense (Schön, 1983) and intuition of the decision-maker are at least equally important tools. Second, decomposition must be practiced with restraint so that the complexity of the interactions among parts is not overshadowed by the much simpler behavior of each of the individual parts. Third, it must be understood that the definition of the parts is largely dependent on the objectives and knowledge about the problem that is currently available to the decision-maker. Even relatively minor discoveries about the greater whole, of which the given problem situation forms a part, are likely to have significant impact on the purpose and the objectives of the problem situation itself.

2.7 Decision-Making in Complex Problem Situations

As shown in Fig. 2.3, there are several characteristics that distinguish a complex problem from a simple problem. First, the problem is likely to involve many related issues or variables. As discussed earlier the relationships among the variables often have more bearing on the problem situation than the variables themselves. Under such tightly coupled conditions it is often not particularly helpful, and may even be misleading, to consider issues in isolation. Second, to confound matters some of the variables may be only partially defined and some may yet to be discovered. In any case, not all of the information that is required for formulating and evaluating alternatives is available. Decisions have to be made on the basis of incomplete information.

Third, complex problem situations are pervaded with dynamic information changes. These changes are related not only to the nature of an individual issue, but also to the context of the problem situation. For example, a change in wind direction during a major brushfire may have a profound impact on the entire nature of the relief operation. Apart from precipitating an immediate re-evaluation of the firefighting strategy, it may require the relocation of firefighters and their equipment, the re-planning of evacuation routes, and possibly even the relocation of distribution centers. Certainly, a change in the

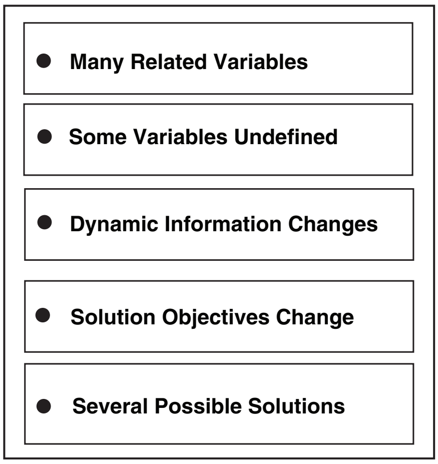


Fig. 2.3. Character of complex problems

single factor of wind direction could, due to its many relationships, call into question the very feasibility of the existing course of action (i.e., the firefighting plan). Even under less critical conditions it is not uncommon for the solution objectives to change several times during the decision-making process. This fourth characteristic of complex problem situations is of particular interest. It exemplifies the tight coupling that can exist among certain problem issues, and the degree to which decision-makers must be willing to accommodate fundamental changes in the information that drives the problem situation.

Fifth, complex problems typically have more than one solution (Archea, 1987). A solution is found to be acceptable if it satisfies certain performance requirements and if it has been determined that the search for alternatives is no longer warranted. Such a determination is often the result of resource constraints (e.g., availability of time, penalty of non-action, or financial resources) rather than a high level of satisfaction with the quality of the proposed solution.

While human decision-making in complex problem situations has so far defied rigorous scientific explanation, we do have knowledge of at least some of the characteristics of the decision-making activity.

- In search of a starting point for assessing the nature of the problem situation, decision makers will typically look for prominent aspects that are likely to have significant impact on the solution space. These aspects or problem features are then used to at least temporarily establish boundaries within which the initial problem solving efforts can be focused. However, the qualifying terms temporarily and initial are appropriately chosen since both the selected features and the early boundaries are likely to change many times due to the highly dynamic nature of the decision-making process.

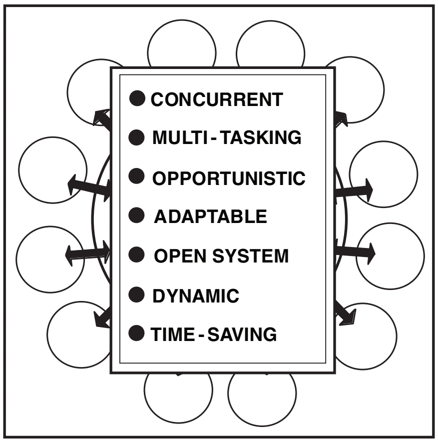


Fig. 2.4. Parallel decision-support

- The complexity of the decision-making activity does not appear to be due to a high level of difficulty in any one area but the multiple relationships that exist among the many issues that impact the desired outcome. Since a decision in one area will tend to influence several other areas there is a need to consider many factors at the same time. This places a severe burden on the human cognitive system. As mentioned previously, although the neurological mechanisms that support conscious thought processes are massively parallel, the operation of these reasoning capabilities is largely sequential. Accordingly, decision-makers tend to apply simplification strategies for reducing the complexity of the problem-solving activity. In this regard it becomes readily apparent why second Wave software with multi-tasking capabilities provides a much more useful architecture for decision-support systems (Fig. 2.4).
- Observation of decision-makers in action has drawn attention to the important role played by experience gained in past similar situations, knowledge acquired in the general course of decision-making practice, and expertise contributed by persons who have detailed specialist knowledge in particular problem areas. The dominant emphasis on experience is confirmation of another fundamental aspect of the decision-making activity. Problem-solvers seldom start from first principles. In most cases, the decision-maker builds on existing solutions from previous situations that are in some way related to the problem under consideration. From this viewpoint, the decision-making activity involves the modification, refinement, enhancement and combination of existing solutions into a new hybrid solution that satisfies the requirements of the given problem system. In other words, problem-solving can be described as a process in which relevant elements of past prototype solution models are progressively and collectively molded

into a new solution model. Very seldom are new prototype solutions created that do not lean heavily on past prototypes.

Finally, there is a distinctly irrational aspect to decision-making in complex problem situations. Schön refers to a "…reflective conversation with the situation…" (Schön, 1983). He argues that decision-makers frequently make value judgments for which they cannot rationally account. Yet, these intuitive judgments often result in conclusions that lead to superior solutions. It would appear that such intuitive capabilities are based on a conceptual understanding of the situation, which allows the problem solver to make knowledge associations at a highly abstract level.

Based on these characteristics the solution of complex problems can be categorized as an information intensive activity that depends for its success largely on the availability of information resources and, in particular, the experience and reasoning skills of the decision-makers. It follows that the quality of the solutions will vary significantly as a function of the problem-solving skills, knowledge, and information resources that can be brought to bear on the solution process. This clearly presents an opportunity for the useful employment of computer-based decision-support systems in which the capabilities of the human decision-maker are complemented with knowledge bases, expert agents, and self-activating conflict identification and monitoring capabilities.

2.8 Principal Elements of Decision-Making

Over the past two decades that our Collaborative Agent Design Research Center has been developing distributed, collaborative decision-support systems some insights have been gained into the nature of the decision-making activity. In particular, we have found it useful to characterize decision-making in terms of six functional elements (Fig. 2.5): *information; representation; visualization; communication; reasoning*; and, *intuition*.

2.8.1 The Information Element

Decision-making in complex problem situations is a collaborative activity involving many sources of information that are often widely dispersed. Seldom is all of the information required for the solution, or even only a component of the problem, physically located in the immediate vicinity of the decision-maker. In fact, much of the information is likely to reside in remote repositories that can be accessed only through electronic means, the telephone, e-mail, or the temporary relocation of a member of the problem-solving team (Fig. 2.6). If the desired information requires expert advice the services of a consultant may be required in addition to, or instead of, access to an information resource.

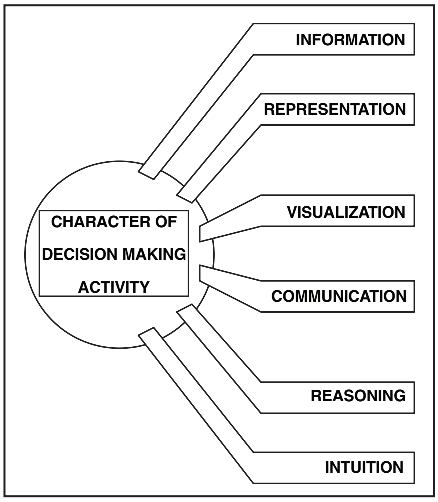


Fig. 2.5. Decision-making elements

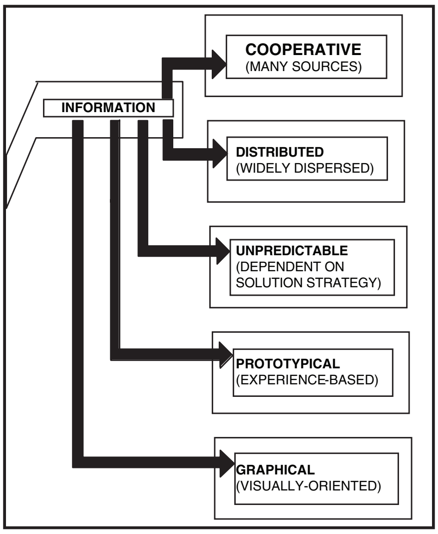


Fig. 2.6. The *information* element

The term information is used here in the broadest sense to include not only factual data and the progressively more comprehensive and detailed description of the problem system, but also the many knowledge bases that are part of the local and global environment within which the problem situation is constituted. In this regard, we are concerned with the knowledge of the individual members of the problem-solving team, the knowledge of peripheral players (e.g., colleagues, associates and consultants), the collective knowledge of the profession (such as the various engineering professions, the military establishment, or the management profession) and industry, and beyond that those aspects of what might be referred to as global knowledge that impact the problem context.

Typically, the problem specifications (i.e., constraints, criteria, and objectives) evolve with the problem solution as the decision-makers interact with the problem situation. Accordingly, the information requirements of the problem solver are not predictable since the information needed to solve the problem depends largely on the solution strategy adopted (Fischer and Nakakoji, 1991). In this respect problem solving is a learning process in which the decision-maker progressively develops a clearer understanding of the problem that is required to be solved. Much of the information that decision-makers use in the development of a problem solution is gleaned from experience with past projects. In fact, it can be argued that solutions commonly evolve out of the adaptation, refinement and combination of prototypes (Gero, 1988). This argument suggests that the more expert human decision-makers are the more they tend to rely on prototypical information in the solution of complex problems. It would appear that the accumulation, categorization and ability to apply prototype knowledge are the fundamental requirements for a human decision-maker to reach the level of expert in a particular domain. Based largely on the work of Gero (1988) and Rosenman and Gero (1993) the following techniques used by engineering designers to develop solutions through the manipulation of prototypes can be identified as being universally applicable to other problem domains:

- *Refinement.* The prototype can be applied after changes have been made in the values of parameter variables only (i.e., the instance of the prototype is reinterpreted within the acceptable range of the parameter variables).
- *Adaptation.* Application of the prototype requires changes in the parameters that constitute the description of the prototype instance, based on factors that are internal to the prototype (i.e., a new prototype instance is produced).
- *Combination.* Application of the prototype requires the importation of parameter variables of other prototypes, producing a new instance of a reinterpreted version of the original prototype.
- *Mutation.* Application of the prototype requires structural changes to the parameter variables, either through internal manipulations or the importation of parameter variables from external sources (i.e., either a reinterpreted version of the original prototype or a new prototype is produced).
- *Analogy.* Creation of a new prototype based on a prototype that exists in another context, but displays behavioral properties that appear to be analogous to the application context.

For application purposes in knowledge-based decision-support systems prototypes may be categorized into five main groups based on knowledge content (Schön, 1988; Pohl and Myers, 1994):

1. Vertical prototype knowledge bases that contain typical object descriptions and relationships for a complete problem situation or component thereof. Such a knowledge base may include all of the types that exist in a particular problem setting, for example: an operational template for a particular kind of humanitarian relief mission; a certain type of propulsion unit; or, a building type such as a library, sports stadium, or supermarket.
2. Horizontal prototype knowledge bases that contain typical solutions for sub-problems such as commercial procurement practices, construction of a temporary shelter, or techniques for repairing equipment. This kind of knowledge often applies to more than one discipline. For example, the techniques for repairing a truck apply equally to the military as they do to auto-repair shops, engineering concerns, and transportation related organizations.
3. Domain prototype knowledge bases that contain guidelines for developing solutions within contributing narrow domains. For example, the range of structural solutions appropriate for the construction of a 30-story office building in Los Angeles is greatly influenced by the seismic character of that region. Posed with this design problem structural engineers will immediately draw upon a set of rules that guide the design activity. Similarly, an acoustic consultant engaged to determine the cause of noise transmission between two adjacent office spaces will diagnose the problem based on experience with previous situations. The possibility of the existence of indirect sound transmission paths, such as a false ceiling, is likely to receive early consideration.
4. Exemplar prototype knowledge bases that describe a specific instance of an object type or solution to a sub-problem. Exemplary prototypes can be instances of vertical or horizontal prototypes, such as a particular building type or a method of welding a certain kind of steel joint that is applied across several disciplines and industries (e.g., building industry and automobile industry). Decision-makers often refer to exemplary prototypes in exploring solution alternatives to sub-problems.
5. Experiential knowledge bases that represent the factual prescriptions, strategies and solution conventions employed by the decision-maker in solving similar kinds of problem situations. Such knowledge bases are typically rich in methods and procedures. For example, a particularly memorable experience such as the deciding event in a past business negotiation or the experience of seeing for the first time the magnificent sail-like concrete shell walls of the Sydney Opera House, may provide the basis for a solution method that is applied later to create a similar experience in a new problem situation that may be quite different in most other respects. In other words, experiential prototypes are not bound to a specific type of

problem situation. Instead, they represent techniques and methods that can be reproduced in various contexts with similar results. Experiential knowledge is often applied in very subtle ways to guide the solution of subproblems (e.g., a subterfuge in business merger or take-over negotiations that is designed to mislead a competing party).

The amount of prototypical information is potentially overwhelming. However, the more astute and experienced decision-maker will insist on taking time to assimilate as much information as possible into the problem setting before committing to a solution theme. There is a fear that early committal to a particular solution concept might overlook characteristics of the problem situation that could gain in importance in later stages, when the solution has become too rigid to adapt to desirable changes. This reluctance to come to closure places a major information management burden on the problem solver. Much of the information cannot be specifically structured and prepared for ready access, because the needs of the problem solver cannot be fully anticipated. Every step toward a solution generates new problems and information needs (Simon, 1981).

One of the early targets of ontology-based multi-agent systems is the integration of existing information sources (i.e., databases and legacy applications) into comprehensive decision-support systems. The initial focus of such efforts is the linkage of existing databases. This is not a trivial task since these existing information resources typically were implemented in many different ways. Consequently, any integrating system will be required to support the conceptual integration of a variety of data resource models. This can be accomplished through the provision of several internal-level database representations, requiring a number of additional mapping functions to link the internal and conceptual representation levels. In this way, any externally linked database can be removed or replaced by another database, simply by recoding the internal to conceptual level mapping. Since this will not affect the external data representation, the user-interfaces built at the external level will also remain unchanged.

The scope of database query facilities desirable for the kind of multi-agent, decision-support environment discussed here far exceeds traditional database management system (DBMS) functions. They presuppose a level of embedded intelligence that has not been available in the past. Some of these desirable features include: conceptual searches instead of factual searches; automatically generated search strategies instead of predetermined search commands; multiple database access instead of single database access; analyzed search results instead of direct (i.e., raw) search results; and, automatic query generation instead of requested searches only.

A traditional DBMS typically supports only factual searches. In other words, users and applications must be able to define precisely and without ambiguity what data they require. In complex problem situations users rarely know exactly what information they require. Often they can define in only conceptual terms the kind of information that they are seeking. Also, they would like to be able to rely on the DBMS to automatically broaden the search with a view to discovering information.

This suggests, in the first instance, that an intelligent DBMS should be able to formulate search strategies based on incomplete definitions. It should be able to infer, from rather vague information requests and its own knowledge of the requester and the problem context, a set of executable query procedures. To facilitate this process the DBMS should maintain a history of past information requests, the directed search protocols that it generated in response to these requests, and at least some measure of the relative success of the previous search operations.

A traditional DBMS normally provides access to only a single database. A knowledge-based decision-support environment is likely to involve many information sources, housed in a heterogeneous mixture of distributed databases. Therefore, through the internal-level database representations discussed earlier, the DBMS must be able to access multiple databases. Using the mapping functions that link these internal representations an intelligent DBMS should be capable of formulating the mechanisms required to retrieve the desired data from each source, even though the internal data structures of the sources may differ widely. Particularly when search results are derived from multiple sources and the query requests themselves are vague and conceptual in nature, there is a need for the retrieved information to be reviewed and evaluated before it is presented to the requester. This type of search response formulation facility has not been necessary in a traditional DBMS, where users are required to adhere to predetermined query protocols that are restricted to a single database.

Finally, all of these capabilities (i.e., conceptual searches, dynamic query generation, multiple database access, and search response formulation) must be able to be initiated not only by the user but also by any of the computer-based agents that are currently participating in the decision-making environment. These agents may be involved in any number of tasks that require the import of additional information from external databases into their individual knowledge domains.

A conceptual model of an intelligent DBMS interface (Fig. 2.7) with the capabilities described above should be able to support the following typical information search scenario that might occur in an integrated and distributed, collaborative, multi-agent, decision-support environment. Queries that are formulated either by the user or generated automatically by a computer-based agent are channeled to a Search Strategy Generator. The latter will query a Search Scenario Database to determine whether an appropriate search strategy already exists from a previous search. If not, a new search strategy is generated, and also stored in the Search Scenarios Database for future use. The search strategy is sent to the Database Structure Interpreter, which automatically formulates access protocols to all databases that will be involved in the proposed search. The required access and protocol information, together

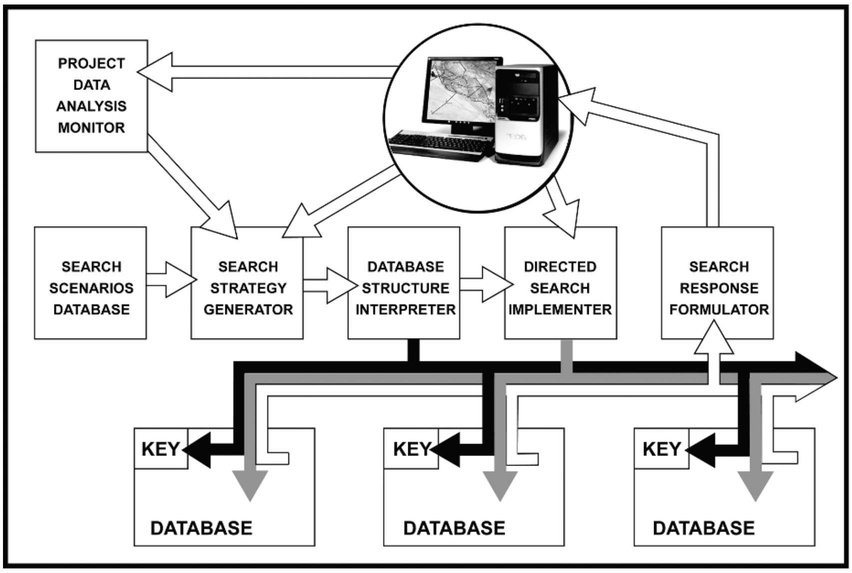


Fig. 2.7. Conceptual model of an intelligent DBMS interface

with the search strategy, are sent to the Directed Search Implementer, which conducts the required database searches. The results of the search are sent to a Research Response Formulator, where the raw search results are analyzed, evaluated and combined into an intelligent response to be returned to the originator of the query.

The proposition that the DBMS interface should be able to deal with incomplete search requests warrants further discussion. When searching for information, partial matching is often better than no response. In traditional query systems, a database record either matches a query or it does not. A flexible query system, such as the human brain, can handle inexact queries and provide best guesses and a degree of confidence for how well the available information matches the query (Pohl et al., 1994). For example, let us assume that a military commander is searching for a means of trapping a given enemy force in a particular sector of the battlefield and formulates a something like a choke point query. In a flexible query system a something like operator would provide the opportunity to match in a partial sense, such as: terrain conditions that slow down the movement of troops; unexpected physical obstacles that require the enemy to abruptly change direction; subterfuge that causes enemy confusion; and so on. These conditions can all, to varying extent, represent something like a choke point that would be validated by a degree of match qualification.

Flexible query processing systems are fairly common. For example, most automated library systems have some level of subject searching by partial keyword or words allowing users to browse through a variety of related topics.

Even word-processing programs include spelling checkers, which by their very nature search for similar or related spellings. However, even a flexible query system cannot automatically form hypotheses, since the system does not know what to ask for.

The ability to search for something like is only a starting point. How can the system be prompted to search for vaguely or conceptually related information? For example, how can the system discover the intuitive connection between a physical choke point, such as a narrow cross-corridor in a mountainous battlefield, and a precision fire maneuver aimed at concentrating enemy forces in an exposed area? In other words, how can the system show the commander that the precision fire maneuver option can satisfy the same intent as the cross-corridor option? In addition, the system must not overwhelm the commander with an unmanageable number of such intuitive speculations. To discover knowledge it is necessary to: form a hypothesis; generate some queries; view and analyze the results; perhaps modify the hypothesis and generate new queries; and, repeat this cycle until a pattern emerges. This pattern may then provide insight and advice for intuitive searches. The goal is to automate this process with a discovery facility that repeatedly queries the prototype knowledge bases and monitors the reactions and information utilized by the decision-maker, until the required knowledge is discovered.

2.8.2 The Representation Element

The methods and procedures that decision-makers utilize to solve complex problems rely heavily on their ability to identify, understand and manipulate objects (Fig. 2.8). In this respect, objects are complex symbols that convey meaning by virtue of the explicit and implicit information that they encapsulate within their domain. For example, architects develop design solutions by

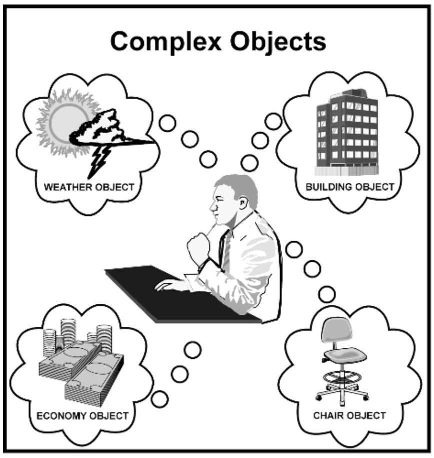


Fig. 2.8. Symbolic reasoning with objects

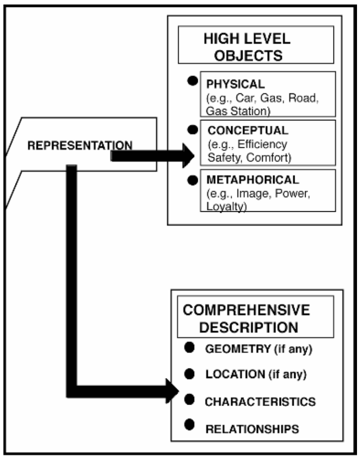


Fig. 2.9. The *representation* element

reasoning about neighborhoods, site characteristics, buildings, floors, spaces, walls, windows, doors, and so on. Each of these objects encapsulates knowledge about its own nature, its relationships with other objects, its behavior within a given environment, what it requires to meet its own performance objectives, and how it might be manipulated by the decision-maker within a given problem scenario (Fig. 2.9). This knowledge is contained in the various representational forms of the object as factual data, relationships, algorithms, rules, exemplar solutions, and prototypes.

The reliance on object representations in reasoning endeavors is deeply rooted in the innately associative nature of the human cognitive system. Information is stored in long-term memory through an indexing system that relies heavily on the forging of association paths. These paths relate not only information that collectively describes the meaning of symbols such as building, car, chair, and tree, but also connect one symbol to another. The symbols themselves are not restricted to the representation of physical objects, but also serve as concept builders. They provide a means for grouping and associating large bodies of information under a single conceptual metaphor. In fact, Lakoff and Johnson (Lakoff and Johnson, 1980) argue that "...our ordinary conceptual system, in terms of which we both think and act, is fundamentally metaphorical in nature." They refer to the influence of various types of metaphorical concepts, such as "...desirable is up" (i.e., spatial metaphors) and "...fight inflation" (i.e., ontological or human experience metaphors), as the way human beings select and communicate strategies for dealing with everyday events.

Problem-solvers typically intertwine the factually based aspects of objects with the less precise, but implicitly richer language of metaphorical concepts. This leads to the spontaneous linkage of essentially different objects through the process of analogy. In other words, the decision-maker recognizes similarities between two or more sub-components of apparently unrelated objects and embarks upon an exploration of the discovered object seeking analogies where they may or may not exist. At times these seemingly frivolous pursuits lead to surprising and useful solutions of the problem at hand.

The need for a high level representation is fundamental to all computer-based decision-support systems. It is an essential prerequisite for embedding AI in such systems, and forms the basis of any meaningful communication between user and computer. Without a high level representation facility the abilities of the computer to assist the human decision maker are confined to the performance of menial tasks, such as the automatic retrieval and storage of data or the computation of mathematically defined quantities. While even those tasks may be highly productive they cannot support a partnership in which human users and computer-based systems collaborate in a meaningful and intelligent manner in the solution of complex problems.

The term high level representation refers to the ability of computer software to process and interpret changes in data within an appropriate context. It is fundamental to the distinction between data-centric and information-centric software. Strictly speaking data are numbers and words without relationships.[2] Software that incorporates an internal representation of data only is often referred to as data-centric software. Although the data may be represented as objects the absence of relationships to define the functional purpose of the data inhibits the inclusion of meaningful and reliable automatic reasoning capabilities. Data-centric software, therefore, must largely rely on predefined solutions to predetermined problems, and has little (if any) scope for adapting to real world problems in near real-time.

Information, on the other hand, refers to the combination of data with relationships to provide adequate context for the interpretation of the data. The richer the relationships the more context and the greater the opportunity for automatic reasoning by software agents. Software that incorporates an internal information model (i.e., ontology) consisting of objects, their characteristics, and the relationships among those objects is often referred to as information-centric software (Pohl et al., 2005). The information model provides a virtual representation of the real world domain under consideration.

For example, let us assume that we wish to represent a component of a building such as a conference room in the computer. Until recently, in a data-centric software environment, we would have treated the conference room as a three-dimensional geometric entity that can be described in terms of points (i.e., x-y-z coordinates), lines, or surfaces. While this may be satisfactory for

[2] Even though data are often stored in a relational database management system, the relationships that are stored with the data in such a database are structural in nature and do not provide any information on how the data will be used (i.e., the *context* of the data)

displaying different internal views of the building space and even generating animated walk-through sequences, it does not provide a basis for the computer to reason about any aspect of the space, such as that a conference room must have a door for it to be usable. To provide the computer with such a reasoning capability the particular entity, in this case the conference room must be represented in the computer as an information structure that constitutes the context of a building. This can be achieved quite easily by storing in the computer the word building and associating this word with some characteristics such as: physical object; made of material; has height, width and length; consists of one or more floors; has spaces on floors; and so on. Then further defining spaces with characteristics such as: enclosed by walls, floor and ceiling; with wall having at least one opening referred to as a door; and so on.

In such an information-centric software environment the same conference room would be presented to and stored in the computer as part of the building information structure (i.e., ontology) to support the following kind of interaction with the user:

Computer user:	I would like to represent a component of a building.
Computer software:	Loads its stored building ontology into memory.
	Asks user: "What kind of a building component?"
Computer user:	A space of type conference.
Computer software:	For how many persons?
Computer user:	Up to 16 persons.
Computer software:	Suggested space size is: 16 ft (length), 14 ft (width), 8 ft (height).
	Suggested furniture: 6 ft by 3 ft table, 16 chairs, screen, white board.
	Other features: There must be at least one door.

As can be seen from the interaction between the user and the computer software, by virtue of the information structure the computer has some understanding of the meaning of a building within the context of its characteristics and the relationships of its components (i.e., floors, spaces, walls, openings, and furniture). This endows the computer software with the ability to collaborate and assist the user by reasoning about the relationships between the data entered by the user and the context contained in the simple information representation provided by the building ontology. Accordingly, driven by the context provided in the ontology, software agents are able to spontaneously reason about the characteristics of a conference room for 16 persons. Beyond establishing the need for at least one exit and the kind of furniture normally required, this could easily extend to the evaluation of the impact on equipment and functionality of an external window.

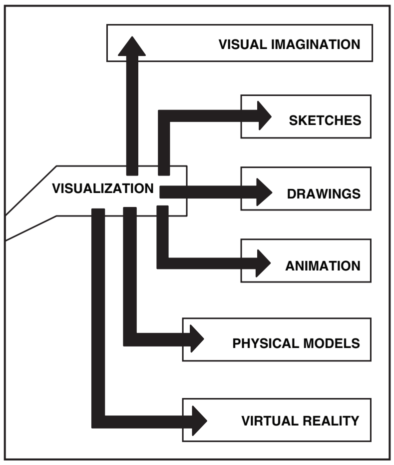


Fig. 2.10. The visualization element

2.8.3 The Visualization Element

Problem solvers use various visualization media, such as visual imagination, drawings and physical models, to communicate the current state of the evolving solution to themselves and to others (Fig. 2.10). Drawings, in particular, have become intrinsically associated with problem solving. Although the decision-maker can reason about complex problems solely through mental processes, drawings and related physical images are useful and convenient for extending those processes. The failings of the drawing as a vehicle for communicating the full intent of the decision-maker do not apply to the creator of the drawing. To the latter the drawing serves not only as an extension of short-term memory, but also as a visual bridge to the associative indexing structure of long-term memory. In this way, every meaningful part of the drawing is linked to related data and deliberation sequences that together provide an effectively integrated and comprehensive representation of the artifact.

From a technical point of view a great deal of headway has been made over the past two decades in the area of computer-based visualization. However, without high-level representation capabilities even the most sophisticated computer generated images are nothing but hollow shells. If the computer system does not have even the simplest understanding of the nature of the objects that are contained in the image then it cannot contribute in any way to the analysis of those objects. On the other hand, visualization in combination with high-level representation becomes the most powerful element of the user-interface of a decision-support system. Under these circumstances, visualization promotes the required level of understanding between the user and the computer as they collaborate in the solution of a problem.

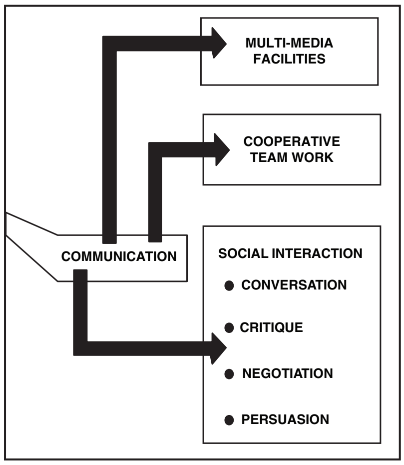


Fig. 2.11. The communication element

2.8.4 The Communication Element

The solution of complex problems is typically undertaken by a team of decision-makers. Each team member contributes within a collaborative decision-making environment that relies heavily on the normal modes of social interaction, such as conversation, critique, negotiation, and persuasion (Fig. 2.11). Two aspects of such an interactive environment are particularly well catered for in computer-based systems. The first aspect relates to the ability of computer-driven communication networks to link together electronically based resources located anywhere on Earth or in space. Technical advances in the communication industry have greatly enhanced the ability of individuals to gain access to remotely distributed information sources, and to interact with each other over vast distances. In fact, connectivity rather than geographical distance has become the principal determinant of communication.

In this respect, the notion of presence is being decisively redefined in an information society. In recent years we have seen the gradual acceptance of a new concept of presence that does not require the physical relocation of persons. Major sporting events and entertainment shows are more conveniently viewed on television from the home. Typically, in the case of sporting events, the quality of the televised presentation of the competition is greatly improved by technical enhancements such as instant replays and close-up shots of particularly skillful maneuvers, explanations and analyses by informed commentators, and short profile films of the best competitors.

Electronic mail, Internet chat groups, telephone and video conferencing facilities, and facsimile (FAX) transmissions, have reduced the need for face-to-face meetings. Commercial companies are gradually being forced to reassess the need for a centralized workplace. Why pay the considerable overhead costs associated with maintaining office space for employees, if the employees could equally well perform their work at home? Computer-based messaging services and global connectivity have already reached a level of reliability and convenience that is more than adequate for business communications.

The second aspect is interwoven with the first by a major advance in the telecommunication industry that occurred some 20 years ago and allowed all types of data to be converted into digital form. Through the use of digital switching facilities modern communication networks are able to transmit telephone conversations and graphical images in the same way as data streams have been sent from one computer to another over the past 40 years.

As a direct result of these advances in communication systems the convenient and timely interaction of all of the members of a widely dispersed problem-solving team is technically assured. It is now incumbent on software developers to produce computer-based decision-support systems that can fully support collaborative teamwork, which is neither geographically nor operationally limited. Such systems will integrate not only computer-based information resources and software agents, but also multiple human agents (i.e., the users) who will collaborate with the computer-based resources in a near real-time, interactive, distributed environment.

2.8.5 The Reasoning Element

Reasoning is central to any decision-making activity. It is the ability to draw deductions and inferences from information within a problem-solving context. The ability of the problem solver to reason effectively depends as much on the availability of information, as it does on an appropriately high level form of object representation (Fig. 2.12). Decision-makers typically define complex problems in terms of issues that are known to impact the desired outcome. The relative importance of these issues and their relationships to each other change dynamically during the decision-making process. So also do the boundaries of the problem space and the goals and objectives of the desired outcome. In other words, the solution of complex problems is an altogether dynamic process in which both the rules that govern the process and the required properties of the end-result are subject to continuous review, refinement and amendment (Reitman, 1964; Reitman, 1965; Rittel and Webber, 1984).

As discussed previously, the complexity of a problem is normally not due to a high degree of difficulty in any one area but the multiple relationships that exist among the many issues that impact the desired outcome. Since a decision in one area will tend to influence several other areas there is a critical need for concurrency. However, the reasoning capabilities of the human problem solver are sequential in nature. Accordingly, decision-makers find it exceedingly difficult to consider more than three or four issues at any one time. In an attempt to deal with the concurrency requirement several strategies are

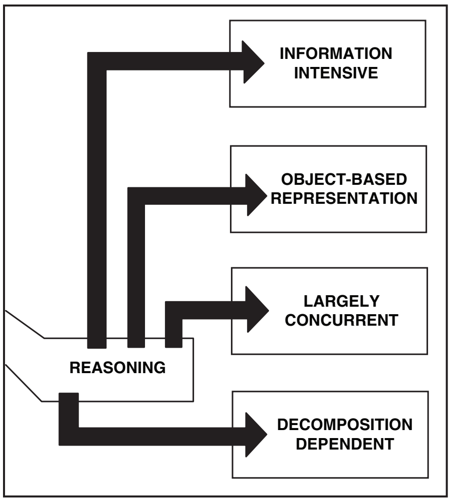


Fig. 2.12. The reasoning element

commonly employed to reduce the complexity of the reasoning process to a manageable level.[3]

- *Constraint identification.* By sifting through the available information the problem-solver hopes to find overriding restrictions and limitations that will eliminate knowledge areas from immediate consideration.
- *Decision factor weighting.* By comparing and evaluating important problem issues in logical groupings, relative to a set of predetermined solution objectives, the decision-maker hopes to identify a smaller number of issues or factors that appear to have greater impact on the final solution. Again, the strategy is to reduce the size of the information base by early elimination of apparently less important considerations.
- *Solution conceptualization.* By adopting early in the decision-making process a conceptual solution, the problem-solver is able to pursue a selective evaluation of the available information. Typically, the problem-solver proceeds to subdivide the decision factors into two groups, those that are compatible with the conceptual solution and those that are in conflict. By a process of trial and error, often at a superficial level, the problem-solver develops, adapts, modifies, re-conceives, rejects and, often, forces the preconceived concept into a final solution.

In complex problem situations reasoning proceeds in an iterative fashion through a cycle of analysis, synthesis and evaluation (Fig. 2.13). During

[3] Reasoning is a logical process that proceeds in a step-by-step manner. In this respect reasoning is quite different from intuition, which allows humans to spontaneously come to conclusions that are neither consciously formulated nor explainable at the time of their first appearance

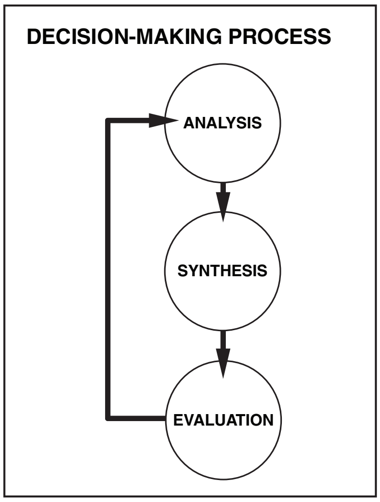


Fig. 2.13. Reasoning methodology

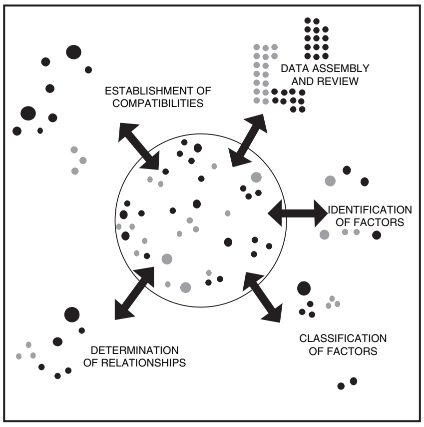


Fig. 2.14. Analysis stage of reasoning

the analysis stage (Fig. 2.14) the problem-solver interprets and categorizes information to establish the relative importance of issues and to identify compatibilities and incompatibilities among the factors that drive these issues.

During synthesis (Fig. 2.15) solution boundaries and objectives are continuously reexamined as the decision-maker develops narrow solutions to sub-problems and combines these narrow solutions into broader solutions. Initially, these solution attempts are nothing more than trial balloons. Or, stated in more technical terms, explorations based on the development of the relationships among the principal issues and compatible factors identified during the analysis stage. Later, as the problem-solving activity progresses,

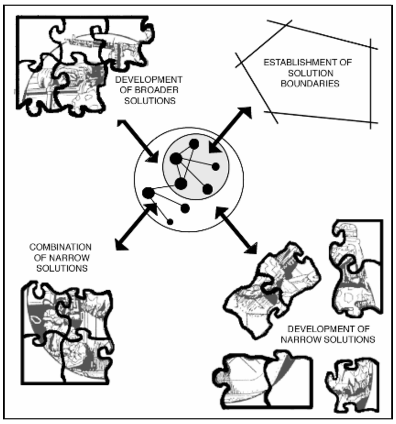


Fig. 2.15. Synthesis stage of reasoning

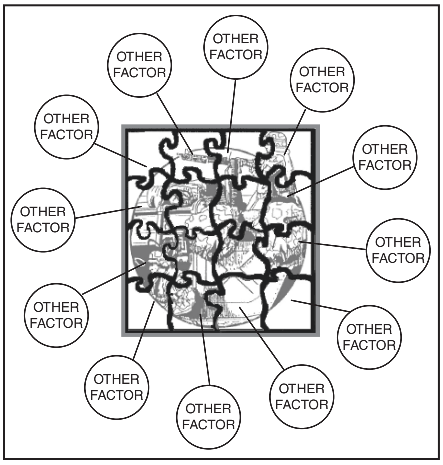


Fig. 2.16. Evaluation stage of reasoning

firmer conceptual solution strategies with broader implications emerge. However, even during later cycles the solution strategies tend to be based on a limited number of issues or factors.

During the evaluation stage (Fig. 2.16) the decision-makers are forced to test the current solution strategy with all of the known problem issues, some of which may have been considered only superficially or not at all during the formulation of the current solution proposal. This may require the current solution concepts to be modified, extended or altogether replaced. Typically, several solution strategies are possible and none are completely satisfactory. Archea (1987), in his description of the architectural design activity refers to this activity as "…puzzle-making", suggesting by implication that the decision-maker utilizes the reasoning cycle more as a method for exploring the problem space than as a decision-making tool for forcing an early solution.

2.8.6 The Intuition Element

Schön (1983, 1988), has written extensively about the intuitive aspects of decision-making. Although he focused primarily on engineering design as an application area, his views provide valuable insight into the solution of complex problems in general. Design has all of the common characteristics of complex problem situations, and some additional ones such as the desire for solution uniqueness in architectural design, that make it a prime candidate for computer-based assistance (Pohl et al., 1994).

In Schön's (1988) view designers enter into "... design worlds" in which they find the objects, rules and prototype knowledge that they apply to the design problem under consideration. The implication is that the designer continuously moves in and out of design worlds that are triggered by internal and external stimuli. While the reasoning process employed by the designer in any particular design world is typically sequential and explicitly logical, the transitions from state to state are governed by deeper physiological and psychological causes. Some of these causes can be explained in terms of associations that the designer perceives between an aspect or element of the current state of the design solution and prototype knowledge that the designer has accumulated through experience. Others may be related to emotional states or environmental stimuli, or interactions of both (Fig. 2.17).

For example, applying Schön's view to the broader area of complex problem solving, a particular aspect of a problem situation may lead to associations in the decision-maker's mind that are logically unrelated to the problem under

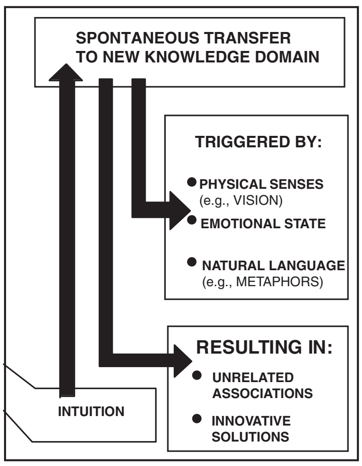


Fig. 2.17. The *intuition* element

consideration. However, when the decision-maker pursues and further develops these associations they sometimes lead to unexpected solutions. Typically, the validity of these solutions becomes apparent only after the fact and not while they are being developed. In popular terms we often refer to these solutions as creative leaps and label the author as a brilliant strategist. What we easily forget is that many of these intuitions remain unrelated associations and do not lead to any worthwhile result. Nevertheless, the intuitive aspect of decision-making is most important. Even if only a very small percentage of these intuitive associations lead to a useful solution, they still constitute one of the most highly valued decision-making resources.

The reasons for this are twofold. First, the time at which the decision-maker is most willing to entertain intuitive associations normally coincides with the most difficult stage in the problem solving process. Typically, it occurs when an impasse has been reached and no acceptable solution strategy can be found. Under these circumstances intuition may be the only remaining course of action open to the decision-maker. The second reason is particularly relevant if there is a strong competitive element present in the problem situation, for example, during a chess game or during the execution of military operations. Under these circumstances, strategies and solutions triggered by intuitive associations will inevitably introduce an element of surprise that is likely to disadvantage the adversary.

The importance of the intuition element itself in decision-making would be sufficient reason to insist on the inclusion of the human decision-maker as an active participant in any computer-based decision system. In designing and developing such systems in our Center over the past two decades we have come to appreciate the importance of the human–computer partnership concept, as opposed to automation. Whereas in some of our early systems (e.g., ICADS (Pohl et al., 1998) and AEDOT (Pohl et al., 1992)) we included agents that automatically resolve conflicts, today we are increasingly moving away from automatic conflict resolution to conflict detection and explanation. We believe that even apparently mundane conflict situations should be brought to the attention of the human agent. Although the latter may do nothing more than agree with the solution proposed by the computer-based agents, he or she has the opportunity to bring other knowledge to bear on the situation and thereby influence the final determination.

2.9 The Human–Computer Partnership

To look upon decision-support systems as partnerships between users and computers, in preference to automation, appears to be a sound approach for at least two reasons. First, the ability of the computer-based components to interact with the user overcomes many of the difficulties, such as representation and the validation of knowledge, that continue to plague the field of machine learning (Forsyth, 1989; Thornton, 1992; Johnson-Laird, 1993).

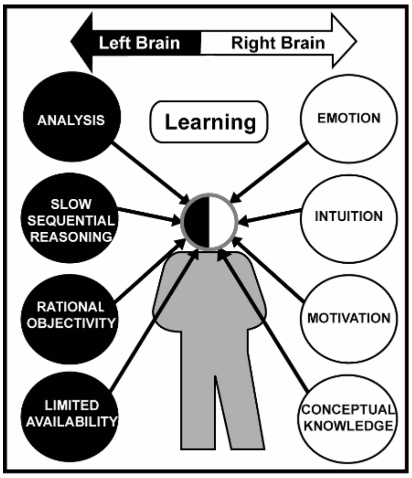


Fig. 2.18. Human abilities and limitations

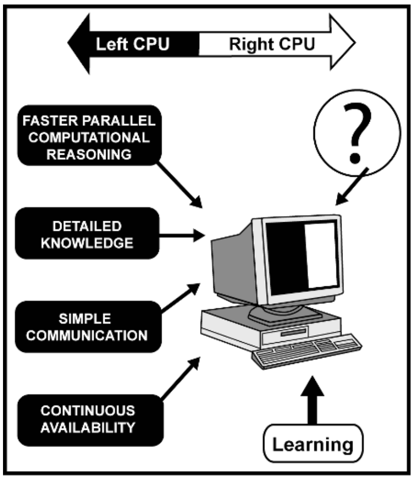


Fig. 2.19. Computer abilities and limitations

Second, human and computer capabilities are in many respects complementary (Figs. 2.18 and 2.19). Human capabilities are particularly strong in areas such as communication, symbolic reasoning, conceptualization, learning, and intuition (Fig. 2.18). We are able to store and adapt experience and quickly grasp the overall picture of even fairly chaotic situations. Our ability to match patterns is applicable not only to visual stimuli but also to abstract concepts and intuitive notions. However, as powerful as these capabilities may appear to be they are nevertheless flawed by those innately human tendencies that were discussed at the beginning of this chapter under the rubric of human limitations and weaknesses.

Decision-making based on analysis requires not only a great deal of rather tedious and time consuming work, but also the unbiased and unemotional evaluation of past experience and possible future outcomes. This is indeed a tall order since emotions are a driving force in virtually all human activities. Pomerol and Adam (2008), in Sect. 2.5 of Chap. 1, discuss in some detail the critical role that emotions play in decision-making. Due to the influence of emotions, coupled with our aversion to hard work, our experience-based cognitive facilities, and our desire for instant gratification, we easily fall prey to over-reliance on intuition. In contrast to the painstaking sequential logic that must be applied in an analytical process, intuition is almost entirely subconscious and produces almost immediate results rather effortlessly. However, intuition can easily lead to false conclusions (Bonabeau, 2003). Unfortunately, we often see patterns that do not exist in reality. The greater the complexity of a situation the more likely that our intuitive assessments may be incorrect, since we are so easily misled by our almost uncontrollable desire to maintain the status quo. As a result we tend to judge new circumstances in the context of past conditions, particularly when these judgments are made intuitively. Examples of such human failings have been provided by Hammond et al. (2002) and include the following types of deceptions:

The anchoring deception. We tend to use the first information received as a reference point for comparing all subsequent information. For example, posing a question in the form of "Is the distance between Chicago and New York greater than 2,000 kilometers?" is likely to produce an answer that is biased toward a distance of 2,000 km. Clearly, intelligent software that operates in partnership with its human user would not only serve as a convenient source of factual data, but also assist the user in viewing a problem from several perspectives by providing access to information sources and reference points. The intelligence of the software is then related to its ability to automatically detect the need for such assistance, rather than the assistance functions themselves.

The status quo deception. We feel most comfortable with current conditions and practices unless compelled to change by threatening events. The tendency to delay decisions is fundamental to common expressions such as "... let's wait until things settle down" or "... I'll rethink this later". In this case, an intelligent decision-support system should be able to not only alert the user to threatening events, but also assist in tracing the historical path that has led to the status quo conditions and undertake a detailed analysis of alternative courses of action.

The confirming evidence deception. We often favor a particular course of action without adequate justification. In such situations we tend to rely heavily on ad hoc confirming evidence, instead of undertaking the necessary analysis. The factual analysis and evaluation capabilities of an intelligent decision-support system are particularly useful as a counterpoint to the bias and relative laziness of the human decision-maker.

The sunken costs deception. We have difficulty admitting to past errors in judgment and may stubbornly insist on the perpetuation of a decision path that is fundamentally flawed. Under these circumstances the evidence generated through the analysis, evaluation and consequence determination capabilities of an intelligent software partner may be the only effective method of exposing the deception.

The forecasting deception. We can easily become overconfident without corroborating experience or too prudent by relying on a worst case scenario. In this case intelligent decision-support software can assist through the very nature of its disciplined approach to problem solving. It is in this area that the human decision-maker is particularly vulnerable because the absence of experience with future events will force reliance on past experience that may only partially, or not at all, apply.

The framing trap. A poorly framed problem can easily bias our decisions since we tend to be unduly influenced by risks associated with potential losses, even if these risks are remote. The absence of emotions in intelligent decision-support systems for the foreseeable future should be helpful in this regard. They allow the decision-maker to consider a problem from several different reference points. However, care must be taken by the designers of the software to ensure that the results of the computer-based analysis are presented to the human user in a neutral manner, so that potential gains and losses are more likely to be considered on an equal basis.

As these examples indicate, intelligent software systems can be particularly helpful in complementing human capabilities by providing a tireless, fast and emotionless problem analysis and solution evaluation capability. Large volumes of information and multi-faceted decision contexts tend to easily overwhelm human decision-makers. When such an overload occurs we tend to switch from an analysis mode to an intuitive mode in which we have to rely almost entirely on our ability to develop situation awareness through abstraction and conceptualization. While this is perhaps our greatest strength it is also potentially our greatest weakness, because at this intuitive meta-level we become increasingly vulnerable to emotional influences.

The capabilities of the computer are strongest in the areas of parallelism, speed and accuracy (Fig. 2.19). Whereas the human being tends to limit the amount of detailed knowledge by continuously abstracting information to a higher level of understanding, the computer excels in its almost unlimited capacity for storing data. While the human being is prone to impatience, loss of concentration and panic under overwhelming or threatening circumstances, the computer is totally oblivious to such emotional influences. The most effective implementation of these complementing human and machine capabilities is in a tightly coupled partnership environment that encourages and supports seamless interaction.

In conclusion, it is certainly appropriate to revisit Kurzweil's hypothesis mentioned early on in this chapter that computing devices are a natural ingredient and necessary requirement for accelerating the intellectual evolution of human beings. For this hypothesis to hold true, intelligent software systems would need to be able to compensate for at least three recognized limitations of the human cognitive system; namely: poor performance in the absence of experience; emotional interference with logical processes; and, a distinct lack of motivation for proactive endeavors. Existing computer capabilities that show promise in this regard include: information storage in context building ontological structures; symbolic reasoning; pattern matching; computation speed; virtually unlimited parallelism; low level learning; analogy detection; and, tireless unemotional task performance. However, several additional capabilities would appear to be required. These include at least the following three capabilities that are likely to challenge the developers of AI-based decision-support systems for the next several decades.

First, there is a need for automatic context generation to form the basis of higher level learning capabilities. While much headway has been made during the past two decades in the representation of context using rich information structures such as ontologies, these are still largely static, predefined virtual models of real world knowledge domains. What are needed are methods for extending and merging ontologies dynamically during software execution (i.e., extensible information representation models). Current industry research efforts in this area such as the WebFountain™ project (IBM, 2002; Chase, 2002), are interesting but have not yet led to breakthrough advances in AI.

Second, there is a need for an interface to seamlessly link intelligent software with the human nervous system. Currently available interface devices and virtual reality capabilities are still very primitive. While some very promising advances have been made in the bio-engineering field in recent years with implanted sensors for artificial limbs, artificial hearing devices, and the invasive monitoring of bodily functions, much more progress needs to be made before we can contemplate the feasibility of a practical implanted user-interface.

Third, there is a need for new methodologies that will allow the development of software that can support the creation of knowledge through analogous reasoning or other as yet unknown processes. The notion of a conceptual database search, discussed previously in the context of the *information element* of the decision-making process (Fig. 2.6), is an example of such a capability. The realization of this kind of AI capability is likely to be the furthest goal to reach.

References

Archea J., Puzzle-Making: what architects do when no one is looking, in Kalay (ed.) Principles of Computer-Aided Design: Computability of Design, Wiley, New York, NY (1987)

Bohm D., Wholeness and the Implicate Order, Ark Paperbacks, London, England (1983)

Bonabeau E., Dont Trust Your Gut, Harvard Business Review, May (2003)
Brockman J., The singularity: a talk with ray kurzweil, Edge 99 – March 25, 2002, Edge Foundation (2002) (http://www.edge.org)
Brooks R., Flesh and Machines: How Robots Will Change Us, Pantheon Books, Random House, New York, NY (2002)
Chase V., Made to Order: IBM Makes Sense of Unstructured Data, Think Research Web Site, http://www.research.ibm.com/thinkresearch/pages/2002/2002/0308_unstructured.shtml (2002)
DOE, Research Abstracts from the DOE Human Genome Program: Contractor-Grantee Workshop VIII, US Dept. of Energy, February 27 to March 2, Santa Fe, New Mexico (2000) (http://www.ornl.gov/hgmis/publicat/00santa)
Finn R., Neural prosthetics come of age as research continues, The Scientist, 11/19 (1997) 13–16
Fischer G. and K. Nakakoji, Making design objects relevant to the task at hand, Proc. AAAI-91 Ninth National Conference on Artificial Intelligence, MIT Press, Cambridge, MA (1991) 67–73
Forsyth R., Machine Learning: Principles and Techniques, Chapman & Hall, New York, NY (1989)
Gero J., M. Maher and W. Zhang, Chunking Structural Design Knowledge as Prototypes, Working Paper, The Architectural Computing Unit, Department of Architectural and Design Science, University of Sydney, Sydney, Australia (1988)
Hammond J., R. Keeney and H. Raiffa, The Hidden Traps in Decision Making, Harvard Business Review, September–October (1998)
Hübener M., D. Shoham, A. Grinvald and T. Bonhoeffer, Spatial relationships among three columnar systems in Cat Area 17, Journal of Neuroscience, 17 (1997) 9270–9284
IBM, WebFountain™, http://www.almaden.ibm.com/webfountain/(2002)
Johnson-Laird P., Human and Machine Thinking, Erlbaum, Hillsdale, NJ (1993)
Kelly K., Out of control: the new biology of machines, Social Systems and The Economic World, Perseus Books, Reading, MA (1994) 298–302
Kiernan V., Brains at work: researchers use new techniques to study how humans think, Chronicle of Higher Education, 44/20 (1998) A16–17
Klein, G. Sources of Power: How People Make Decisions, MIT Press, Cambridge, MA (1998)
Kurzweil R., The Age of the Spiritual Machines, Viking, Penguin Group, New York, NY (1999)
Lakoff G. and M. Johnson, Metaphors We Live By, University of Chicago Press, Chicago, IL (1980)
Minsky M., Logical vs. analogical or symbolic vs. connectionist or neat vs. scruffy, in Artificial Intelligence at MIT, Expanding Frontiers, Patrick H. Winston (ed.), Vol. 1, MIT Press, Cambridge, MA (1990) 218–243
Pohl J., Some notions of complex adaptive systems and their relationship to our world, Intersymp-99, in Advances in Collaborative Decision-Support Systems for Design, Planning and Execution, Baden-Baden, Germany, August 2–7 (1999)
Pohl J., Some Thoughts on Human Nature: A Discussion of Human Strengths and Weaknesses, InterSymp-2002, Focus Symposium on Collaborative Decision-Support Systems, Baden-Baden, Germany, July 29 to August 30, Preconference Proceedings (June 2002)

Pohl J., Intelligent software systems in historical context, in Jain L. and G. Wren (eds.), Decision Support Systems in Agent-Based Intelligent Environments, Knowledge-Based Intelligent Engineering Systems Series, Advanced Knowledge International (AKI), Sydney, Australia (2005)

Pohl J., Elements of Human Decision-Making, Proceedings InterSymp-2006, Advances in Computer Cybernetics, Vol. XIII, The International Institute for Advanced Studies in Systems Research and Cybernetics, Baden-Baden, Germany, August 7–12 (2006)

Pohl J. and L. Myers, A distributed cooperative model for architectural design, in G. Carrara and Y. Kalay (eds.), Knowledge-Based Computer-Aided Architectural Design, Elsevier, Amsterdam (1994) 205–242

Pohl J., A. Chapman, L. Chirica, R. Howell and L. Myers, Implementation Strategies for a Prototype ICADS Working Model, Technical Report, CADRU-02–88, CAD Research Center, Design and Construction Institute, College of Architecture and Environmental Design, Cal Poly, San Luis Obispo, CA (1988)

Pohl J., J. La Porta, K. Pohl and J. Snyder, AEDOT Prototype (1.1): An Implementation of the ICADS Model, Technical Report, CADRU-07–92, CAD Research Center, Design and Construction Institute, College of Architecture and Environmental Design, Cal Poly, San Luis Obispo, CA (1992)

Pohl J., L. Myers and A. Chapman, Thoughts on the Evolution of Computer-Assisted Design, Technical Report, CADRU-09–94, CAD Research Center, Design and Construction Institute, College of Architecture and Environmental Design, Cal Poly, San Luis Obispo, CA, September (1994)

Pohl J., A. Chapman, K. Pohl, J. Primrose and A Wozniak, Decision-Support Systems: Notions, Prototypes, and In-Use Applications, Collaborative Agent Design Research Center, Technical Report CADRU-11–97, Cal Poly, San Luis Obispo, CA 93407 (1997) 10–11

Pomerol J.C. and F. Adam, Understanding human decision making – a fundamental step towards effective intelligent design support, in G. Phillips – Wren, L.C. Jain (eds.), Intelligent Decision Making: An AI-Based Approach, Chapter I (2008)

Powledge T., Unlocking the secrets of the brain, Bioscience, 47 (two-part article) July (1997) 330–334 and August (1997) 403–409

Reitman W., Heuristic decision procedures, open constraints, and the ill-defined problems, in Shelley and Bryan (eds.) Human Judgments and Optimality, Wiley, New York, NY (1964) 282–315

Reitman W., Cognition and Thought, Wiley, New York, NY (1965)

Rittel H. and M. Webber, Planning problems are wicked problems, in cross (ed.) Developments in Design Methodology, Wiley, New York, NY (1984) 135–144

Rosenman M. and J. Gero, Creativity in design using design prototypes, in Gero and Maher (eds.) Modeling Creativity and Knowledge-Based Creative Design, Erlbaum, Hillsdale, NJ (1993) 111–138

Schön D., The Reflective Practitioner: How Professionals Think in Action, Basic Books, London (1983)

Schön D., Designing: rules, types and worlds, Design Studies, 9(3), (1988) 181–190

Senge P., Transforming the Practice of Management, Human Resource Development Quarterly, Jossey-Bass, San Francisco, CA (1993)

Simon H., The Sciences of the Artificial, MIT Press, Cambridge, MA (1981)

Thornton C., Techniques in Computational Learning, Chapman & Hall, Computing Series, London, England (1992)

Part II

Methods: Computational Intelligence

3

Introduction to Computational Intelligence for Decision Making

Witold Pedrycz[1], Nikhil Ichalkaranje[2], Gloria Phillips-Wren[3], and Lakhmi Jain[4]

[1] Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada
[2] Research Statistics and Technology Branch, Department of Communications, IT and the Arts, GPO Box 2154, CANBERRA ACT 2601
[3] The Sellinger School of Business and Management, Loyola College in Maryland, Baltimore, MD 21210, USA
[4] School of Electrical and Information Engineering, Knowledge Based Intelligent Engineering Systems Centre, University of South Australia, Mawson Lakes, SA 5095, Australia

Summary. Computational intelligence techniques are increasingly extending and enriching decision support through such means as coordinating data delivery, analyzing data trends, providing forecasts, ensuring data consistency, quantifying uncertainty, anticipating the user's data needs, providing information to the user in the most appropriate forms, and suggesting courses of action. This chapter provides an introduction to computational intelligence to enhance decision making.

3.1 Introduction

A refined class of computational intelligence techniques is revolutionalizing the support of decision making, especially under uncertain conditions, by such means as coordinating data delivery, analyzing data trends, providing forecasts, ensuring data consistency, quantifying uncertainty, anticipating the user's data needs, providing information to the user in the most appropriate forms, and suggesting courses of action. This chapter provides an introduction to computational intelligence techniques and applications that can support decision making. Other chapters in the book explore research associated with advances in methods such as neural networks, evolutionary computing and intelligent agents that can be utilized in decision making support.

Computational intelligence paradigms are used to mimic the behavior of humans in some limited yet meaningful manner. These include tools such as symbolic logic, artificial neural networks (ANNs), evolutionary computing, intelligent agents and probabilistic reasoning models (Jain and De Wilde, 2001; Jain and Martin, 1999). In conventional programming methodologies,

W. Pedrycz et al.: *Introduction to Computational Intelligence for Decision Making*, Studies in Computational Intelligence (SCI) **97**, 79–96 (2008)
www.springerlink.com

explicit logic and numerical calculations are provided to solve a problem. In contrast, an ANN mimics some biological systems by solving problems using training and learning to generalize for new problems.

Uncertain and imprecise knowledge can be represented with fuzzy logic (Jain, 1995) and ANNs (Hammerstrom, 1993). They are effective ways of describing complex behavior that is difficult to describe mathematically using conventional methods. Evolutionary computing techniques (Jain and Martin, 1999) evolve a solution to a problem guided by algorithms such as optimization of a multi-dimensional problem. A widely reported category of evolutionary algorithm is a genetic algorithm (GA).

Computational intelligence paradigms have been used successfully to solve problems in many disciplines including business, management, engineering design, medical diagnosis, decision making and web-based systems (Hammerstrom, 1993; Jain and Jain, 1997; Tonfoni and Jain, 2003; Abraham et al., 2005; Jain et al., 2000, Phillips-Wren and Jain, 2005). One fruitful area of research appears to be the fusing of these paradigms using hybrid agents (Jain and Jain, 1997).

3.2 Computational Intelligence in Decision Making

The application of computational intelligence to decision making is certainly not new. Recent advances have made computational intelligence techniques accessible to a wider audience as seen by the increase in the number of applications in such areas as intelligent decision support systems. Computational intelligence is being used in decision support for tasks such as aiding the decision maker to select actions in real-time and stressful decision problems; reducing information overload, enabling up-to-date information; providing a dynamic response with intelligent agents; enabling communication required for collaborative decisions; and dealing with uncertainty in decision problems. Leading computational intelligence professional organizations recognize the current effort in "focusing on problems, not on hammers. Given that we (i.e. Artificial Intelligence researchers) do have a comprehensive toolbox, issues of architecture and integration emerge as central" (Mackworth, 2005). Several applications are given in later chapters in this book demonstrating the pragmatic applications of various computational intelligence techniques.

Other recent examples include an expert system to automate the operations of petroleum production and separation facilities (Chan, 2005). Such systems provide access to plants in remote areas by automatically collecting, transmitting and analyzing data for analysis. The system is able to monitor operations, detect abnormalities, and suggest actions to the human operator based on domain-specific expertise acquired during development of the system. A preliminary evaluation of the system showed satisfactory results.

Case based reasoning (CBR) is being applied to health services in a variety of areas (Bichindaritz and Marling, 2006). Current application of

CBR is in bioinformatics, support to the elderly and people with disabilities, formalization of CBR in biomedicine, and feature and case mining. Recent advances are design of CBR systems to account for the complexity of biomedicine, to integrate into clinical settings and to communicate and interact with diverse systems and methods.

Collaborative decision making and knowledge exchange can be enabled with Artificial Intelligence even in difficult clinical healthcare decisions by incorporating a social context (Frize et al., 2005). In sophisticated neonatal intensive care units, parents, physicians, nurses and other parties must collaborate to decide whether to initiate, limit, continue or discontinue intensive treatment of an infant. The system integrates likely outcomes of the treatment with the physician's interpretation and parents' perspectives. It provides a method of communicating difficult information in a structured form that is still personalized and customized to facilitate decision making.

Fuzzy modeling incorporated into a decision support system has been used to enable forest fire prevention and protection policies in Greece, although the system can be applied on a global basis (Iliadis, 2005). Existing approaches use specific geographic boundaries in determining long-term forest fire risk. An inference mechanism based on fuzzy sets has been demonstrated to estimate forest fire risk more successfully. Avineri (2003) presents a fuzzy decision support system for the selection of transportation projects. The selection procedure is a multiple objectives process, and projects are rated using linguistic variables on both on a quantitative and qualitative basis. Both fuzzy weighted average and noncompensatory fuzzy decision rules are used to describe a given transportation policy,

An ANN is used by Konar et al. (2003) to develop a scheme for criminal investigation using multi-sensory data including voice, fingerprint, facial image and incidental description. When matching results are poor, the speaker identification scheme RBF-BP Neural Net is invoked. When no conclusion about the suspect could be detected by voice, incidental description is used as the resource for criminal investigation. Kates et al. (2000) present a decision support system for diagnosing breast cancer using neural networks. The authors took into account the time dependence of underlying risk structures in the formulation of the neural network.

Genetic programming has been used in a decision support system for a tactical air combat environment (Abraham et al., 2005). The system uses a combination of unsupervised learning for clustering the data and three well-known genetic programming techniques to classify the different decision regions accurately, namely, Linear genetic programming (LGP), Multi-expression programming (MEP) and gene expression programming (GEP). The clustered data are used as inputs to the genetic programming algorithms.

Intelligent agents (IA) are perhaps the mostly widely used applied artificial intelligence method in recent years. Their utilization has significantly advanced many applications, particularly Web-based systems (see for example, Phillips-Wren and Jain, 2005). Learning can be incorporated into agent characteristics to extend the capability of systems (Valluri and Croson, 2005).

The following sections describe briefly a selected number of various computational intelligence paradigms used in decision making.

3.2.1 Neural Networks as a Conceptual and Computing Framework of Decision Making

Neural networks and neurocomputing, in general, offer a comprehensive computational framework inspired by mechanisms of neural sciences and brain functioning which are rooted in learning instead of any pre-programmed behavior. In this sense, neurocomputing becomes fundamentally different from the paradigm of hardwired, instruction-based models of optimization. Artificial neural networks (neural networks, for short) exhibit some characteristics of biological neural networks in the sense the constructed networks include some components of distributed representation and processing. The generalization capabilities of neural networks form one of their most outstanding features. The ability of neural networks to generalize, viz. develop solutions that are meaningful beyond the scope of the learning data, is commonly exploited in various applications.

From the architectural standpoint, a neural network consists of a collection of simple nonlinear processing components called neurons, which are combined together via a net of adjustable numeric connections. A typical mathematical model of a single neuron (Anthony and Bartlet, 1999) comes in a form of an n-input single-output nonlinear mapping of the form of a nonlinear transformation of some weighted sum of its inputs, that is $y = f(\sum_{i=1}^{n} w_i x_i)$ where $x_1, x_2, \ldots, x_n$ are the inputs of the neuron while $w_1, w_2, \ldots, w_n$ are the associated connections (weights).

Neural network structures are characterized by the connection patterns that link the neurons arranged in layers. There are two generic topologies of neural networks, namely feedforward and recurrent (feedback) networks. Feedforward neural networks can exhibit a single layer of neurons or could come as multilayer structures. An example of a single layer network is illustrated in Fig. 3.1 while Fig. 3.2 shows a three-layer network. In general, we may envision multilayer topologies of neural networks.

Recurrent neural networks distinguish themselves from feedforward networks by admitting feedback loops as outlined in Fig. 3.3. These networks may or may not have hidden layers. Recurrent neural networks can exhibit full or partial feedback, depending on how the individual feedback loops have been structured.

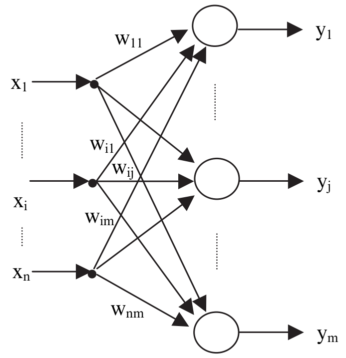


Fig. 3.1. A two-layer architecture of a feedforward neural network

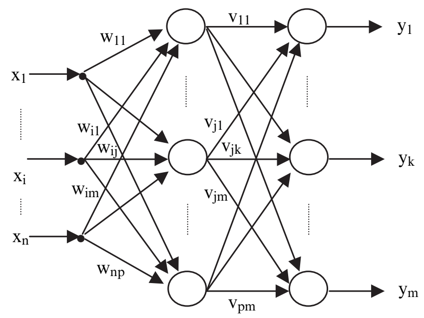


Fig. 3.2. An example of a three-layer feedforward neural network

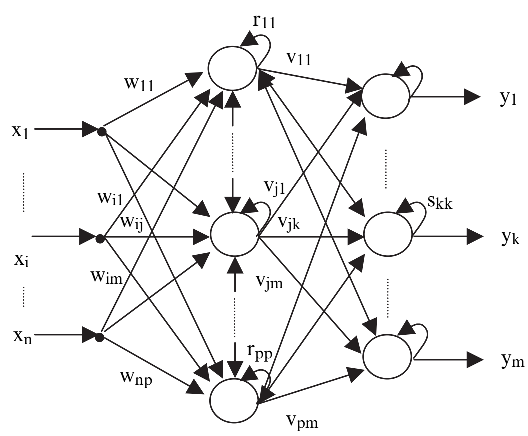


Fig. 3.3. An example of a recurrent neural network

As far as the representation capabilities of neural networks are concerned, they are expressed in the form of a so-called theorem of universal approximation. This theorem states that a feedforward network with a single hidden layer (where the neurons in this layer are equipped with sigmoid type of transfer function) and an output layer composed of linear neurons (viz. with linear transfer functions) is a universal approximator (Cybenko, 1989; Hornik, 1989; 1993). In other words, there exists a neural network of such topology that can approximate any given bounded continuous function $\mathbf{R}^{n} \rightarrow \mathbf{R}$ to any arbitrarily *small* approximation error.

The theorem about universal approximation of neural networks has to be put in a certain context. It is definitely an important and fundamental finding since it assures us about the potential representation capabilities of neural networks. This finding is a typical existence theorem since it does not offer any constructive clue on how such a neural network could be constructed. Theoretical foundations on approximation of functions completed in any metric space as well as algorithmic issues are addressed in Courrieu (2005).

There are three main learning strategies in neural networks, namely (a) supervised, (b) unsupervised, and (c) reinforcement learning. In supervised learning, the network is provided with a training set, pairs of inputs and the corresponding outputs samples. Weights are adjusted in such a way that we construct the network to produce outputs that are as close as possible to the known outputs (targets) of the training set. Unsupervised learning does not require any outputs associated with the corresponding input. The objective of this learning is to reveal the underlying structure existing in the data (e.g., correlations or associations between patterns in data leading to emergence of their possible categories). Reinforcement learning concerns learning processes in which the network receives only high-level guidance as the correctness of its behavior (for instance, we offer a numeric assessment of performance of the network over a collection of some temporal data rather than each data point individually). Learning regarded as an optimization process exhibits two facets, that is, parametric and structural learning. Parametric learning concerns adjustments of the numeric values of the connections. Structural learning involves an optimization of the structure (topology) of the network. Hybrid learning combines supervised and unsupervised learning; here a subset of weights is updated using supervised learning, while some other parameters could be formed through a process of unsupervised learning.

Neural networks cut across an extremely broad spectrum of disciplines and application areas. In decision-making, we can envision a number of fundamental scenarios that are used (Saridakis and Dentsoras, 2006; Chen and Lin, 2003; Azadeh et al. (in press); Gholamian et al., 2006). The capabilities of universal approximation are essential to the formation of a nonlinear mapping between a collection of objectives (say goals and constraints) and resulting decisions. Given the fact that there is a finite training set of pairs of objectives and associated decisions, a neural network forms a nonlinear mapping that links objectives with the corresponding decisions. As usual, we need to maintain a sound balance between approximation capabilities (accuracy of mapping) of the network and its generalization aspects.

While a neural network constitutes suitable computing machinery endowed with substantial learning capabilities, one should stress that the issues of representation of data (objectives and decisions) are equally important. While such data could be numeric, numerous decision processes could be inherently associated with uncertainty or granularity of information we might encounter in decision processes. Under such circumstances, both objectives and decisions could be conveniently modeled as information granules. In particular, we could encounter fuzzy sets of objectives and fuzzy sets of decisions. A way of representing them as fuzzy sets is realized in several ways depending upon the character of fuzzy sets. For instance, we may involve some parametric representation of fuzzy sets considering that all of them are of the same type, say having triangular membership functions. Instead of a single numeric value of a certain objective, we consider a triple of numbers representing lower, modal, and upper bound of the triangular fuzzy set. In the case of Gaussian-like fuzzy sets, we can view a two-parameter representation space in which we involve modal values and spreads of the fuzzy sets. In a nutshell, we envision some interesting symbiotic linkages between neural networks and fuzzy sets with fuzzy sets playing a role of a useful and human-centric interface, refer to Fig. 3.4 which highlights an essence of the resulting topologies.

Quite commonly it is stressed that neural networks are "black boxes" meaning that while approximation capabilities are provided, the resulting network is difficult to interpret. One should be aware that the high level of distributed processing prevents us from bringing the interpretability of the networks to the picture. The interpretability, on the other hand, would be highly beneficial at least for two reasons: (a) it would help us articulate the mapping from objectives and decisions being made thus making the resulting decision process highly transparent, (b) it would be advantageous to accommodate some initial domain knowledge (say, some initial decision rules) as a part of the neural network. In this way, we may anticipate that further learning of such network could be accelerated and made far more efficient.

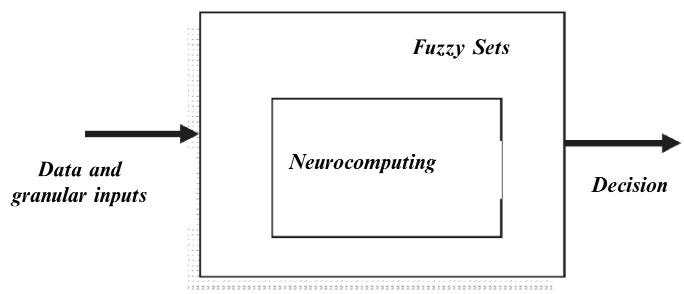


Fig. 3.4. A synergistic neurofuzzy setup of fuzzy sets and neural networks in decision-making problems

A certain category of neural networks which meet these requirements come under the name of fuzzy logic networks (Pedrycz, 1993; Pedrycz and Gomide, 1998). The crux of the constructs lies in the logic – driven processing realized by the individual neurons applied there. In general, we distinguish between two main categories of neurons:

(a) *Aggregative neurons*, typically referred to as AND and OR neurons. Their role is to carry out logic (*and*-like or *or*-like) aggregation of inputs. The connections of the neurons are essential for learning purposes. They allow us to model different impacts associated with various inputs (viz. objectives).
(b) *Referential neurons.* As the name stipulates, these neurons support referential or predicate-based processing in which we are provided with flexibility of expressing relational constraints (*less than, greater than, similar, different*) where each of these two-argument predicates is articulated in the language of fuzzy logic (so the predicates can be satisfied to some extent). Depending upon the predicate being used, the pertinent neurons are referred to as SIM (similarity), INCL (inclusion), DOM (dominance) and alike.

Fuzzy logic networks help us substantially alleviate the shortcomings we have highlighted above. First, as the architecture is highly transparent, any initial domain knowledge could be easily deployed on the network and its topology could be made reflective of the available knowledge hints. By the same token, the network, once trained, could be converted into a logic-based description of relationships. Furthermore, owing to the numeric values of the connections, these relationships are numerically quantified. In this sense, fuzzy logic networks offer an interesting and valuable capability of developing a highly adaptive and interpretable structure with the enriched paradigm of learning on a basis of data and knowledge hints, Fig. 3.5.

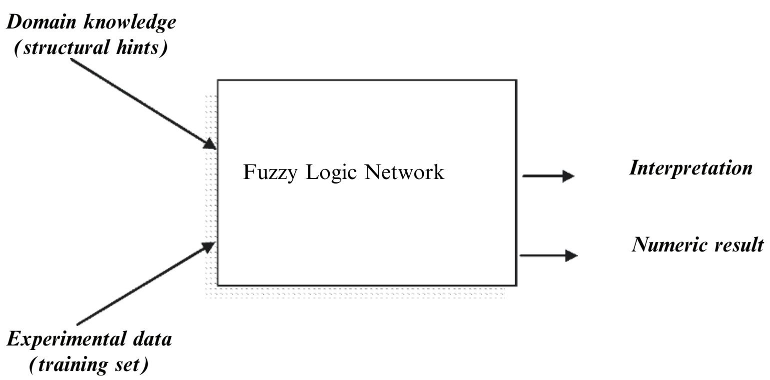


Fig. 3.5. Learning and interpretation of fuzzy logic networks

3.2.2 Evolutionary Computing in Decision Making

To fully benefit from the potential of advanced models of decision-making, there is a genuine need for exploiting effective mechanisms of their *global* optimization. Biologically inspired optimization offers a wealth of optimization mechanisms that tend to fulfill these essential needs. The underlying principles of these algorithms relate to the biologically motivated schemes of system emergence, survival, and refinement. Quite commonly we refer to the suite of these techniques as evolutionary computing to directly emphasize the inspiring role of various mechanisms encountered in nature that are also considered as pillars of the methodology and algorithms. The most visible feature of most, if not all such algorithms, is that in their optimization pursuits they rely on a collection of individuals which interact between themselves in the synchronization of joint activities of finding solutions. They communicate between themselves by exchanging their local findings. They are also influenced by each other.

Evolutionary Optimization

Evolutionary optimization offers a comprehensive optimization environment in which we encounter a stochastic search that mimics natural phenomena of genetic inheritance and Darwinian strife for survival. The objective of evolutionary optimization is to find a maximum of a certain objective function "f" defined in some search space **E**. Ideally, we are interested in the determination of a global maximum of "f".

A Population-Based Optimization Principle of Evolutionary Computing

The crux of the evolutionary optimization process lies in the use of a finite population of N individuals (represented as elements of the search space **E**) whose evolution in the search space leads to an optimal solution. Population-based optimization is an outstanding feature of evolutionary optimization and is practically present in all its variants that we can encounter today. The population is initialized randomly (at the beginning of the search process, say, $t = 0$). For each individual we compute its fitness value. This fitness is related to the maximized objective function. The higher the value of the fitness, the more suitable is the corresponding individual as a potential solution to the problem. The population of individuals in **E** undergoes a series of generations in which we apply some evolutionary operators and through them improve the fitness of the individuals. Those of the highest fitness become more profoundly visible by increasing chances to survive and occur in the next generation.

In a very schematic and abstract way, a computing skeleton of evolutionary optimization can be described as follows

INITIALIZE(population)
evaluate population

repeat
  select individuals for reproduction
    apply evolutionary operators
  evaluate offsprings
  replace some old individuals by offsprings
until termination condition is true
  return a best individual

Let us briefly elaborate on the main components of evolutionary computing. Evaluation concerns a determination of the fitness of individuals in the population. The ones with high values of fitness have chances to survive and appear in consecutive populations (generations of the evolutionary optimization). The selection of individuals to generate offsprings is based on the values of the fitness function. Depending on the selection criterion (which could be stochastic or deterministic), some individuals could produce several copies of themselves (clones). The stopping criterion may involve the number of generations which is perhaps the simplest alternative available to us. One could also involve the statistics of the fitness of the population; say, no significant changes in the average values of fitness may trigger the termination of the optimization process. There are two essential evolutionary operators whose role is to carry out the search process in $\mathbf{E}$ and make sure that it secures its effectiveness. The operators are applied to the current individuals. Typically these operators are of stochastic nature and their intensity depends on the assumed probabilities. There are two groups of operators. Crossover (recombination) operators involve two or more individuals and give rise to one or more offsprings. In most cases, the crossover operator concerns two parents and leads to two offsprings. Formally, we can view such crossover operator as a mapping of the form $\mathbf{E} \times \mathbf{E} \rightarrow \mathbf{E} \times \mathbf{E}$. The objective of crossover is to assure that the optimization exploits new regions of the search space as the offsprings vary from the parents. The mutation operator affects a single individual by randomly affecting one or several elements of the vector, In essence, it forms a mapping from $\mathbf{E}$ to itself, $\mathbf{E} \rightarrow \mathbf{E}$.

The evolutionary optimization process is transparent: we start with some initial population of individuals and evolve the population by using some evolutionary operators. An illustration of evolutionary optimization is illustrated in Fig. 3.6.

Observe that in successive populations, they start to be more "focused" producing individuals (solutions) of higher fitness. Typically, an average fitness of the population could fluctuate; however, on average, it exhibits higher values over the course of evolution. The best individual (viz. the one with the highest fitness) is retained from population to population so we do not lose the best solution produced so far. This retention of the best individual in the population is referred to as an *elitist* strategy.

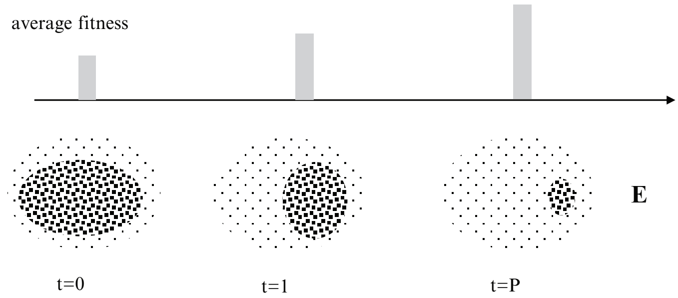


Fig. 3.6. A schematic view at evolutionary optimization; note a more focused populations of individuals over the course of evolution and the increase in fitness values of the individuals and average fitness of the entire population

There are four major categories of evolutionary optimization. While they share underlying principles, they differ in terms of the representation issues and computational aspects.

Evolution strategies (ES) (Schwefel, 1995) are predominantly focused on parametric optimization. In essence, a population consists only of a single individual that is a vector of real numbers. This individual undergoes a Gaussian mutation in which we add a zero mean Gaussian variable of some standard deviation, N(0, σ). The fittest from the parent and the offspring becomes the next parent. The value of the standard deviation is adjusted over the course of evolution. The main operator is mutation. One can also encounter population-based versions of ES, known as (μ + λ)-ES in which μ parents generate λ offsprings.

Evolutionary programming (Fogel et al., 1966) originally focused on evolving finite state machines that were focused on the phenotype space. Similar to ES, there is no initial selection and every individual generates one offspring. Mutation is the evolution operator. The best individuals among parents and offsprings become the parent of the next generation.

Genetic algorithms (GAs) (Holland, 1975; Goldberg, 1989; Michalewicz, 1996) are one of the most visible branches of evolutionary optimization. In its standard format, GAs exploit a binary genotype space {0,1}n. The phenotype could be any space as long as its elements could be encoded into binary strings (bitstrings, for short). The selection scheme is proportional selection, known as the roulette wheel selection. A number of random choices is made in the whole population which implies that an individual is selected with probability that is proportional to its fitness. The crossover operation replaces a segment of bits in the first parent by the corresponding string of the second parent. The mutation concerns a random flipping of the bits. In the replacement, offsprings replace all parents.

Genetic programming (GP) (Koza, 1994; Kinnear, 1994) originated as a vehicle to evolve computer programs, and algebraic and logic expressions, in

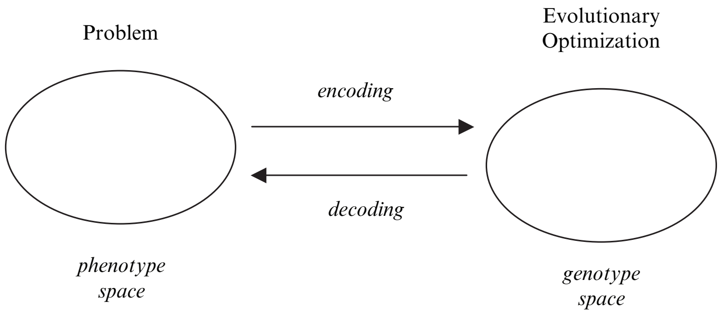


Fig. 3.7. From phenotype space to genotype space: links between optimization problem and its representation in evolutionary optimization

particular. The predominant structures in GP are trees. These are typically implemented in the form of LISP expressions (S-expressions). This realization helped define crossover operation as swapping to sub-trees between two S-expressions as still a valid S-expression).

A suitable problem representation in evolutionary optimization becomes a key issue that predetermines success of the optimization process and implies quality of the produced solution. Let us note that evolutionary optimization is carried out in the *genotype* space **E** which is a result of a transformation of the problem from the original space, a so-called *phenotype* space **P**, realized with the use of some encoding and decoding procedures; refer to Fig. 3.7.

In a more descriptive way, we could view representation issues as being central to the nature of the underlying optimization problem. Knowledge representation is a truly multifaceted problem and as such one has to proceed with prudence realizing that the effectiveness of this scheme implies the quality of evolutionary solution.

In what follows, several examples of encoding and decoding serve as an illustration of the diversity of possible ways of knowledge representation.

1-Binary encoding and decoding: Any parameter assuming real values can be represented in the form of the corresponding binary number. This binary coding is used quite commonly in genetic algorithms (GAs). The strings of bits are then subject to evolutionary operations. The result is decoded into the corresponding decimal equivalent, or more formally, the genotype space, $\mathbf{E} = \{0, 1\}^{\mathrm{m}}$ hypercube where "m" stands for the dimensionality of the space and depends on the number of parameters encoded in this way and a resolution (number of bits) used to complete the encoding.

2-Floating point (real) encoding and decoding: Here we represent values of parameters of the system under optimization using real numbers. Typically, to avoid occurrence of numbers in different ranges, all of them are scaled (e.g., linearly) to the unit intervals so in effect the genotype space is a unit hypercube, $\mathbf{E} = [0, 1]^{\mathrm{p}}$ with "p" denoting the number of parameters. The resulting string of real numbers is re-transformed into the original ranges of the parameters.

Evolutionary optimization offers a number of evident advantages over some other categories of optimization mechanisms. They are general and their conceptual transparency is definitely very much appealing. The population-based style of optimization offers a possibility of a comprehensive exploration of the search space and provides solid assurance of finding a global maximum of the problem. To take full advantage of the potential of evolutionary optimization, one has to exercise prudence in setting up the computing environment. This concerns a number of crucial parameters of the algorithm that concern evolutionary operators, size of population, stopping criterion, to name the most essential ones.

Decision-making models benefit from evolutionary computing in several important ways:

- Structural optimization becomes crucial in many cases as we typically envision a great deal of possible structures of the models. In this sense, evolutionary optimization helps choose an optimal one.
- Decision-making processes typically involve a number of criteria; the multicriterial nature of the problem calls for their simultaneous optimization and here evolutionary techniques become beneficial.
- It is advantageous to view a collection of possible solutions (rather than the optimal one) including those that are suboptimal yet could offer a better insight into the nature of the decision-making problem itself and allow for a global characterization of the solutions.

3.3 Agents in Decision Making

One of the modern Artificial Intelligence (AI) approaches leads towards the most talked-about trend called "intelligent agents". Many researchers believe that agent technology is the result of convergence of many notions and trends within computer science namely AI, cognitive science, object-oriented programming and distributed computing (Wooldridge, 2002; Decker, 2004). The result of such convergence led to the birth of a modern AI field known as distributed artificial intelligence (DAI), which focuses on agents and their "interactions" with environments and peers (multi-agent systems or MAS). In order to simplify the development and study of agent technology, popular categorisation based on agent theories, agent system architectures and agent languages by leading researchers such as Wooldridge (Wooldridge and Jennings, 1995a, b) will help significantly in forming a basic understanding of the agent area. Agent theories define and address reasoning within agents. Agent system architectures facilitate the implementation of agent theories within a specified environment, and agent languages are similar to programming languages which facilitate the theories and system architecture to construct and compile (Wooldridge and Jennings, 1995a, b).

Technological constraints such as the agent's ability to include and interact with humans by exhibiting human social norms such as learning, trust and respecting human privacy need to be addressed in its core theories. In order to address technological constraints one needs to focus on agent theories and ways to include social norms. Learning or adaptation is an important step forward to complement intelligence. Traditional AI notions such as machine learning and cognitive science theories could be of great help to facilitate such social attributes in current agent theories.

Recent popular trends in agent technology include teaming, and adaptation or learning. In the DAI community, the most common term used to describe multi-agent interaction is "teaming". Teaming broadly covers MAS interaction and its resultant child attributes such as *communication* and *coordination*, along with sub-notions such as *cooperation* and *collaboration* with peers, which we describe as *teaming agreements* utilising communication and coordination. We present these notions briefly in the following paragraphs.

Communication. Agents have to communicate in order to convey their intentions. Communication is an integral part of interaction but does not have to be direct. It can be indirect by means of a resulting action. Communication in MAS can be implemented either as message passing or using shared variables (Ehlert, 2001). A variety of protocols exist for agent communication based on agent knowledge manipulation, that is, naturalistic human-like communication. Amongst these, those of significance are knowledge query and manipulation language (KQML) and knowledge interchange format (KIF) (The ARPA Sponsored Knowledge Sharing Effort, 2006), and FIPA's (Foundation of Intelligent Physical Agents) agent communications language (ACL) (FIPA, 2006). Such research contributions in agent communication are close to reaching a standardised state.

Coordination. Coordination is crucial as a means of organising agents, their resources and tasks and thus improving agent performance and resolving conflicts. Ehlert (Ehlert and Rothkrantz, 2001) discusses a simple way of managing coordination via task allocation methods. Ehlert classifies task allocations as centralised, distributed, and emergent. In *centralised task allocation*, one central "leader" conducts task distribution either by imposing tasks upon agents (hierarchical) or by trading/brokering tasks. In *distributed task allocation*, each agent attempts to obtain the services it requires from other agents either by sending requests to agents whom it knows have the required services or by sending requests to all agents and accepting the best offer. Distributed task allocation can be separated further in two ways, allocation by *acquaintances* or by *contract net.* Lastly in *emergent* task allocation, which is characteristic of reactive systems, each agent is designed to perform a specific task, therefore no negotiation is necessary. From Ehlert's categorisation, it is evident that two other important attributes arise, namely "*negotiation*" and "*competition*". These attributes may be utilised to coordinate the agent's activities and resources.

Teaming agreements. Sub-notions such as coordination and collaboration are often confused in terms of definitions and implementation. We like to simplify such notions by stating that communication and coordination are parent class attributes and important to any agent who decides to *interact* with other agents. Thus, no matter what teaming agreements one follows, every entity has to communicate and coordinate their resources, goals, and skills to act as a “team”. Teaming agreements such as cooperation and collaboration become child class attributes by utilising communication and coordination, either directly or indirectly. In simple terms, one can take the meaning of cooperation as coexisting with other entities with the obligation to share one or more resources as a part of a coexistence agreement. On the other hand, collaboration is something that encapsulates parts of coexistence with self-induced motivation to share resources and/or skills to achieve a common goal.

Human-centric agents, an answer to early automation pitfalls. Early machine automation and its techniques largely failed to address the human and the human cognitive process (Urlings, 2003; Bratman, 1987, Russel and Norvig, 2006). This was due to the aggressive introduction of automation based on perceived needs and the tools available at that time. Agent technology may aid in such human-centric automation by means of its inherited attributes from cognitive science.

The human-like reasoning and decision making theories such as Belief Desire Intention (BDI) (Rao and Georgeff, 1995) are attractive candidates of agent technology for human-centric automation. These theories could make the agent a stand-alone substitute for the human by replacing him or her. Although these theories exhibit human-like intelligence, they fall short of human interaction abilities. When achieving such human replacement it is imperative that the human should have final “control” along with being able to interact with the agent to regain control in critical situations. The answer to such a trade-off in *control* is human-centric agents. Recent research efforts define this area as *human-agent teaming*. The possibility of considering the human as an equal part of any system and interacting in co-existence would give agent technology the leading edge, which traditional AI systems have failed to give. Human interaction inherits the same issues in MAS interaction with an added focus on naturalistic and proactive communication with the human (Yen et al., 2001) and adaptability. Along with these issues, involving the human brings to the fore new issues such as respecting the social obligations of human society. These social obligations and norms include control, trust, loyalty, and privacy (Tambe et al., 2006).

Learning. The next stage in human-agent teaming would be to demonstrate the adaptive nature of agents. This adaptive nature (learning) will portray the agent as a smart team member especially when dealing with human counterparts. “Learning” is one of the important attributes for allowing the human to “feel” comfortable to communicate, cooperate and adapt to the environment. Modern reasoning models may utilise “learning” to make agents more human-like. Current learning methods have a strong lineage with machine learning from traditional AI. Hybridising such traditional AI techniques with new reasoning models with the help of cognitive science theories and reward-based learning methods can result in making the hybrid model specifically catered to be more human-like. Such a notion is reported in (Sioutis and Ichalkaranje, 2005; Sioutis, 2006), combining existing evolving methods such as reinforcement learning (Michalewicz, 1996) and cognitive science theories such as the observe, orient, decide and act loop (OODA) (Sutton and Barto, 1998) and Rasmussen's decision ladder (Boyd, 2005) into the BDI reasoning model.

3.4 Summary

Computational intelligence paradigms can enhance human decision making through the use of intelligent decision support systems. Methods used in such systems include symbolic logic, ANNs, evolutionary computing, intelligent agents and probabilistic reasoning models. ANNs use training and learning to generalize for new problems. Uncertain and imprecise knowledge can be represented with fuzzy logic and ANNs. Evolutionary computing techniques such as genetic algorithms evolve a solution to a problem guided by algorithms such as optimization of a multi-dimensional problem. One fruitful area of research appears to be the fusing of these paradigms using hybrid agents.

Acknowledgements

We are grateful to the Springer for their permission to reproduce the contents of some of our papers published in Springer's LNAI proceedings.

References

Abraham, A., Grosan, C., Tran, C., Jain, L.C.: A concurrent neural network-genetic programming model for decision support systems. In Proceedings of the Second International Conference on Knowledge Management (2005) 377–384

Anthony, M., Bartlet, P.L.: Neural Network Learning: Theoretical Foundations. Cambridge University Press, Cambridge, 1999

Azadeh, A., Ghaderi, S.F., Anvari, M., Saberi, M., Izadbakhsh, H.: An integrated artificial neural network and fuzzy clustering algorithm for performance assessment of decision making units. Applied Mathematics and Computation **187** (2007) 584–599

Bichindaritz, I., Marling, C.: Case-based reasoning in the health sciences: What's next? Artificial Intelligence in Medicine **36** (2006) 127–135

Boyd, J.R.: The Essence of Winning an losing, at web-address.: http://www.mindsim.com/MindSim/Corporate/OODA.html, Last accessed. 17 December, (2005)

Bratman, M.E. (ed.): Intention, Plans, and Practical Reason. Harvard University Press, Cambridge, MA (1987)

Chan, C.W.: An expert decision support system for monitoring and diagnosis of petroleum production and separation processes. Expert Systems with Applications **29** (2005) 131–143

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

Courrieu, P.: Function approximation in non-Euclidean spaces. Neural Networks **18** (2005) 91–102

Cybenko, G.: Approximation by superposition of a sigmoidal function. Mathematics of Control, Signals, and Systems **2** (1989) 303–314

Decker, K.: A vision for multi-agent systems programming, presented at programming multi-agent systems, First International Workshop, PROMAS 2003, Dastani, M., Dix, J., and El Fallah Seghrouchni, A. (Eds.): Lecture Notes in Artificial Intelligence, vol. 3067, Springer, Berlin Heidelberg New York (2004), 1–17

Ehlert, P., Rothkrantz, L.: Intelligent Agents in an Adaptive Cockpit Environment, Delft University of Technology, Netherlands, Research Report Ref. DKE01-01,Version 0.2 (October 2001)

Fogel, L., Owens, A., Walsh, M.: Artificial Intelligence Through Simulated Evolution. Wiley, New York, NY (1966)

Foundation for Intelligent Physical Agents (Fipa): FIPA Agent Communication Specifications, at Web-Address.: http://www.fipa.org/repository/aclspecs.html, Last accessed. January (2006)

Frize, M., Yang, L., Walker, R.C., O'Connor, A.M.: Conceptual framework of knowledge management for ethical decision-making support in neonatal intensive care. IEEE Transactions on Information Technology in Biomedicine **9** (2005) 205–215

Gholamian, M.R., Fatemi Ghomi, S.M.T., Ghazanfari, M.: A hybrid intelligent system for multiobjective decision making problems. Computers & Industrial Engineering **51** (2006) 26–43

Goldberg, D. Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley, Reading, MA (1989)

Hammerstrom, D.: Neural networks at work. Spectrum **30** (1993) 26–32

Holland, J.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI (1975)

Hornik, K.: Some new results on neural network approximation. Neural Networks **6** (1993) 1069–1071

Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Networks **2** (1989) 359–366

Iliadis, L.S.: A decision support system applying an integrated fuzzy model for long-term forest fire risk estimation. Environmental Modelling and Software **20** (2005) 613–621

Jain, L.C. (ed.): Electronic Technology Directions Towards 2000 **1**, IEEE Computer Society Press, USA (1995)

Jain, L.C., De Wilde, P. (eds.): Practical Applications of Computational Intelligence Techniques. Kluwer, Norwell, MA, USA (2001)

Jain, L.C., Jain, R.K. (eds.): Hybrid Intelligent Engineering Systems. World Scientific Publishing Company, Singapore (1997)

Jain, L.C., Martin, N.M. (eds.): Fusion of Neural Networks, Fuzzy Sets, and Genetic Algorithms. CRC Press LLC, Boca Raton, FL, USA (1999)

Jain, A. et al. (eds.): Artificial Intelligence Techniques in Breast Cancer Diagnosis and Prognosis. World Scientific Publishing Company, Singapore (2000)

Kinnear, K.E.: Advances in Genetic Programming, vol. 1. The MIT Press, USA (1994)

Koza, J.R. Genetic Programming 2. MIT Press, Cambridge, MA (1994)

Mackworth, A.: The Coevolution of AI and AAAI. AI Magazine **26** (2005) 51–52

Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer Verlag, Berlin (1996)

Pedrycz, W.: Fuzzy neural networks and neurocomputations. Fuzzy Sets and Systems **56** (1993) 1–28

Pedrycz, F. Gomide: An Introduction to Fuzzy Sets: Analysis and Design. MIT Press, Cambridge, MA (1998)

Phillips-Wren, G., Jain, L.C. (eds.): Intelligent Decision Support Systems in Agent-Mediated Environments. IOS Press, The Netherlands (2005).

Rao, A.S., Georgeff, M.P.: BDI agents: from theory to practice. Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95), San Francisco, USA (June 1995), 312–319

Russel, S.J., Norvig, P (ed.): Artificial Intelligence: A Modern Approach, 2nd ed.Prentice-Hall, New Jersey (2006)

Saridakis, K.M., Dentsoras, A.J.: Integration of fuzzy logic, genetic algorithms and neural networks in collaborative parametric design. Advanced Engineering Informatics **20** (2006) 379–399

Schwefel, H.-P.: Evolutionsstrategie und numerische Optimierung, PhD Thesis, Technische Universitat Berlin (1995)

Sioutis, C.: Reasoning and Learning for Intelligent Agents, PhD Thesis, University of South Australia, Adelaide (2006)

Sioutis, C., Ichalkaranje, N.: Cognitive hybrid reasoning intelligent system design, presented at Ninth International Conference, on Knowledge-Based Intelligent Information and Engineering Systems, Melbourne, Australia, Springer-Verlag,Germany (September 2005), 838–843

Sutton, R.S., Barto, A.G. (eds.).: Reinforcement Learning: The MIT Press, Cambridge, MA (1998)

Tambe, M., Bowring, E., Pearce, J., Varakantham, P., Scerri, P., Pynadath, D.: Electric Elves: What Went Wrong and Why, AAAI Spring Symposium on "What Went Wrong and Why" (2006)

The ARPA-Sponsored Knowledge Sharing Effort, KQML Specification Document, at Web-Address.: http://www.cs.umbc.edu/kqml/, Last accessed. Jan (2006)

Tonfoni, G., Jain, L.C. (eds.): Innovations in Decision Support Systems. Advanced Knowledge International, Australia (2003)

Urlings, P.: Teaming Human and Machine, PhD Thesis, University of South Australia, Adelaide (2003)

Valluri, A., Croson, D.C.: Agent learning in supplier selection models. Decision Support Systems **39** (2005) 219–240

Wooldridge, M.J. (ed.): An Introduction to Multiagent Systems. Wiley, Chichester (2002).

Wooldridge, M., Jennings, N.R.: Intelligent agents: theory and practice. The Knowledge Engineering Review **10** (2) (1995a) 115–152

Wooldridge, M. Jennings, N.R.: Theories, Architectures, and Languages : A Survey, Intelligent Agents, ECAI-94 Workshop on Agent Theories, Architectures, and Languages, Wooldridge, M., Jennings, N. R. (eds.): Lecture Notes in Artificial Intelligence, vol. 890, Springer-Verlag, Berlin Heidelberg New York (1995b), 1–39

Yen, J., Yin, J., Ioerger, T.R., Miller, M., Xu, D., Volz, R.A.: CAST: collaborative agents for simulating teamwork, Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01), Seattle, WA (August 2001), 1135–1142

4

Collaborative Decision Making Amongst Human and Artificial Beings

Daniel I. Thomas and Ljubo B. Vlacic

Intelligent Control Systems Laboratory, Griffith University, Australia,
Daniel_Thomas@TechnologyOneCorp.com; L.Vlacic@griffith.edu.au

Summary. This chapter provides an overview of collaborative decision making between human and artificial beings. The chapter presents concepts, examples and scenarios that can assist in designing collaborative systems mixing humans and artificial beings as fully equal partners (FEPs). Human and artificial beings are demonstrated performing tasks cooperatively with each other, while being fully replaceable or interchangeable with other beings regardless of their biological or artificial nature. These beings are also not necessarily aware whether his/her/its game partners are human or artificial. This is not to say that FEPs are equal in decision making abilities, but rather that these partners possess an equal communication ability. As a result, a game player is not aware whether his/her game partner is a human or artificial being.

Also outlined is the collaborative process and how this process allows FEPs to collaborate in a structured manner. Once defined, a simple practical example of a collaborative FEPs system is demonstrated: the electronic meeting room. Shown step by step are the processes and values used to arrive at the final outcome, describing in detail how these human and artificial beings collaborate within the electronic meeting room.

Finally, after working through the play scenario and discussing possible future enhancements, some practical domains where collaborative FEPs are applicable in various industries are defined.

By the end of this chapter the reader should have an understanding of the following topics:

- Understanding the concept of human and artificial beings as collaborative fully equal partners.
- Be introduced to the cognitive elements of artificial beings and how these contribute to constructing a FEPs concept.
- Having been walked through a play scenario example of human and artificial being collaboration, will have the necessary resources to create their own play scenarios.
- Be aware of a number of practical applications for collaborative FEPs in industry applications such as Online Training and Education, Human Resources, Project Management, Transportation and Socially Oriented Computer Games for clinical psychology and behavioral studies.

D.I. Thomas and L.B. Vlacic: *Collaborative Decision Making Amongst Human and Artificial Beings*, Studies in Computational Intelligence (SCI) **97**, 97–133 (2008)
www.springerlink.com

4.1 Introduction

With billions of dollars spent annually on computer game entertainment (Beinisch et al., 2005), there is nobody that can contest the fact that the computer games industry is a serious business. Most intriguing about these figures is the rise of massively multiplayer online (MMO) games as a significant game type. According to this OECD report prepared by Beinisch et al., the attracting factor of this game type is its socially-oriented gaming experience.

Given the social aspect of these games, enhancing the social and collaborative experience would increase the attractiveness of MMO games. Interestingly, augmenting the social and collaborative nature of games (as entertainment) can also provide an enhanced learning experience for educational and training games based upon similar concepts.

We propose that one method to augment the social and collaborative nature of educational and training games is by using artificial beings as fully equal partners. In this chapter, we define how human and artificial beings may effectively collaborate with each other in a socially-oriented setting.

In Sect. 1, we define what a collaborative fully equal partner (FEP) is, and how this concept can enhance a computer game based on a social setting followed by Sect. 2 describing the architecture and attributes of a collaborative computer game supporting FEPs.

In Sect. 3, we apply these principles and describe a simple collaborative process based upon the social interactions of the beings within the computer game scenario.

Collaborative FEP concepts provide a compelling collaborative decision-making concept when applied to the various challenges faced within industry. In Sect. 4 we describe some of these possible applications.

By the end of this chapter, it is expected that the reader shall have an understanding of collaborative FEPs, collaborative principles and how to apply these principles in simple group decision-making situations. We see computer games as a setting that enables modeling an embodiment of interactive group decision making and collaborative work environments that may occur within the physical world.

4.2 Humans and Artificial Beings: Fully Equal Partners

There have been many instances in the past where artificial players have controlled an in-game character as a human would (Laird, 2001). In these instances, the artificial player has typically participated as an opponent. In addition, there are many games where simple artificial players have worked as part of a human player's "team" where they interact with these entities through simple commands.

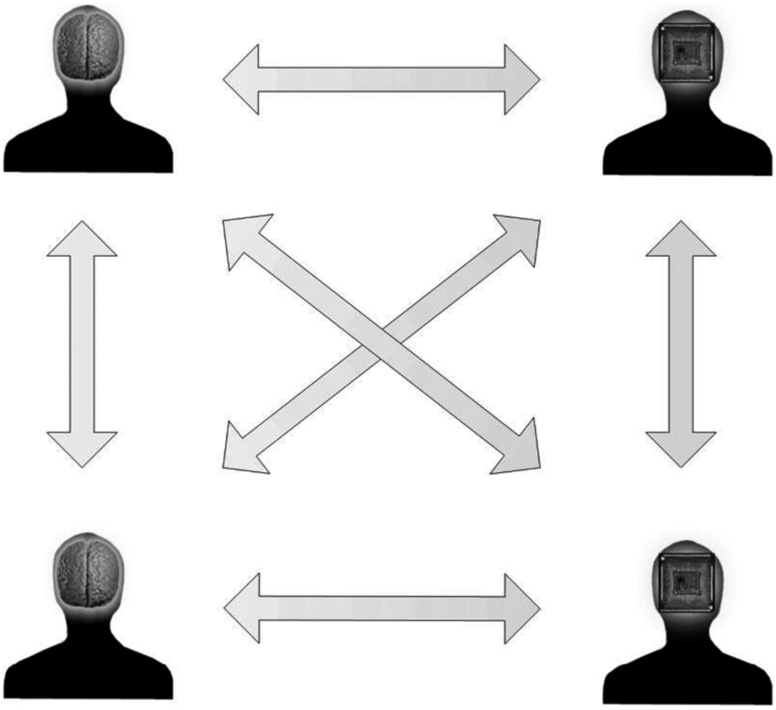


Fig. 4.1. Humans and Artificial beings as collaborative FEPs

Building upon these principles, we consider that if artificial players were able to participate and collaborate within a computer game setting, while having their own internal goals (that is, the ability to play a game as a human would), that these games would have an increased perception of realism and "life" as the interactions between human and artificial beings is not static, scripted or based upon the scenario at hand, but rather changes as these beings interact and collaborate with each other over time to affect change upon the game world that they are situated within.

To this end, we propose a FEP concept (Fig. 4.1) where human and artificial beings collaborate to achieve game goals. We consider this concept as complementary to other uses of autonomous agents as opponents (Laird, 2001) or as interactive story characters (Magerko et al., 2004). Unlike our concept, non-player characters are typically able to work with (or provide simple assistance to) the human players, but do not participate as intelligent collaborative entities, equal in ability to a human being.

A FEP within the context of collaborative computer games:

- Can work cooperatively with other FEP beings (human and artificial) and within the context of computer games;
- "Play" the game as a human would; and

- Does not work to a predefined script or take direction from an agent "director" (Magerko et al., 2004; Riedl et al., 2003).

In addition, collaborative FEPs exhibit the common traits of an autonomous agent. As we consider both human and artificial beings transparently as entities within a collaborative computer game, we find the concept of an agent described by Jennings and Wooldridge (1995) appropriate for application to the characteristics of collaborative FEPs. Therefore, a FEP, being situated within a collaborative computer game enjoys the following abilities:

1. Are Autonomous; operating without the direct intervention of humans or other entities, having control over their internal state.
2. Situated in, and aware of their environment (the game) and are able to interact with this environment through their sensors and effectors.
3. Have some kind of Social Ability; interacting with other human and artificial beings via the use of a communication language.
4. Is able to perceive changes within the (game) environment and react to these changes in a timely fashion.
5. Agents are also Proactive; being able to exhibit goal-directed behavior (taking the initiative) and directly affecting the game and other entities in order to achieve these goals.

Put in more concisely, a FEP (human or artificial) performs tasks cooperatively with other human or artificial beings and is fully replaceable or interchangeable with another FEP. In addition, a being does not know whether his/her/its game partner is a human or artificial being.

4.2.1 Architecture

In our work with collaborative computer games, we see a collaborative computer game architecture consisting of three distinct layers (Fig. 4.2). This layered approach allows us to formalize the necessary attributes required starting from atomic technical concepts through to abstract concepts of the cognitive layer. Since each layer creates an additional abstraction built upon the previous layer, it is important to provide a firm understanding of each layer's function within a collaborative computer game.

We refer to our approach to a layered collaborative computer game architecture for FEPs as the TeamMATE Architecture (Thomas and Vlacic, 2005). Each layer of the TeamMATE Architecture is described in the following sections, demonstrating how this layered approach to collaborative FEPs permits a socially driven environment to exist comprising of human and artificial partners in a heterogeneous relationship.

Before each of the layers of a collaborative FEP system are discussed, it is important to add that in the context of this chapter the term Layer has been used to describe a particular level of the proposed architecture. We believe that the layers put forward here for collaborative computer game architectures

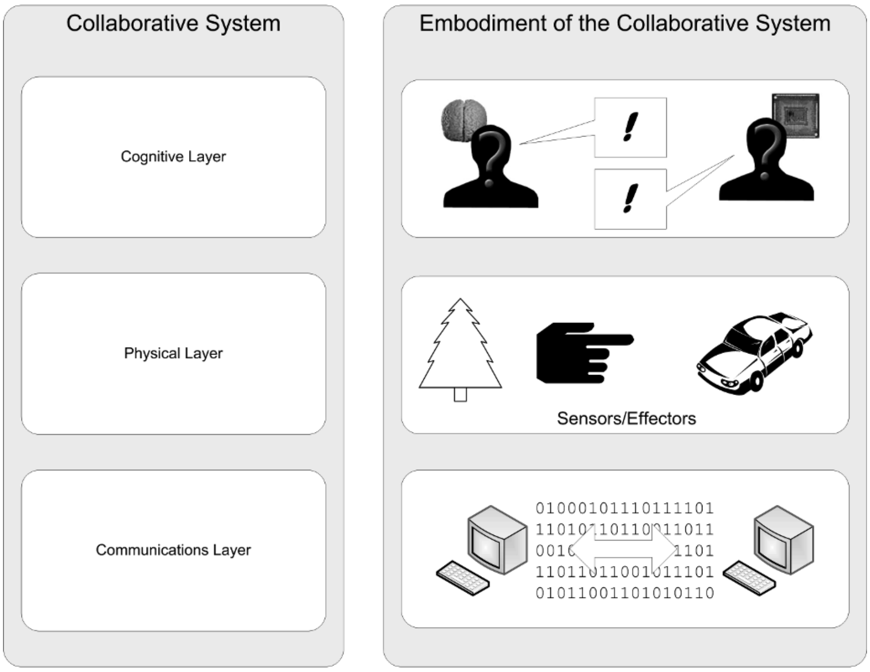


Fig. 4.2. The three architectural layers of a collaborative FEP system

can co-exist with other notions of a layered architecture that deal specifically with the intelligent elements of such a system.

Communication Layer

The communication layer is a very fundamental element of a collaborative computer game. This layer defines the technical protocols used to convey information from the game or other entities from or to the FEP beings situated within the computer game.

The communication layer is effectively a low level transport layer used to pass information from one place to another for example: DirectPlay, TCP/IP, radio signal etc. These protocols, along with the format of the data being transmitted are then available to a FEP's sensors. A FEP may also transmit using these communication protocols via their defined effectors.

Physical Layers

The physical layer within a collaborative computer game defines a FEP's available sensors and effectors within the context of the game. The term "physical" is used to refer to this layer as it defines the characteristics of sensors, effectors and entities within the computer game. Before we are able to work with

Table 4.1. Simplistic physical layer rules

Object	Available actions
Chair	Sit, stand, move
Table	Place item, pick item
Telephone	Call, hang up, speakerphone, listen
Stock	Buy, sell, report
Talk	Whisper, tell all, listen

the more abstract cognitive layers of TeamMATE, it is necessary to define a layer that:

1. Is able to define the physical objects of the collaborative computer game;
2. Provides a common pattern of sensor and effector abilities available to human and artificial beings situated within the game and;
3. Defines the possible actions that may be performed using the available sensors and effectors.

Human and artificial partners must be able to work with the appropriate rules/ constraints of the specific play scenario being undertaken. Physical rules for a given play scenario consist of information about objects in the computer game and how they may be used. Using or enacting some change upon an entity using the defined effectors is referred to as performing an Action. Take as an example, a simple play scenario that contains these physical layer rules (Table 4.1).

When working with more complex games, and also collaborative games that may occur within the physical world, defining all objects and all actions is not feasible. However, it is possible to define the available sensors and effectors for a FEP, while the task of relating objects and actions becomes a function of the cognitive layer.

While a collaborative computer game and the human and artificial beings that are situated within a given play scenario may share a common physical layer, it is not necessarily required that the manifestation of the physical layer will be the same.

For example, in order for a human being to interact with the sensors and effectors provided by the physical layer, it would be necessary to provide a mechanism to interact with the sensors and effectors through a human user interface. Likewise, if an artificial FEP was to interact with other beings within a collaborative computer game, the physical layer would possibly be accessed as some form of software interface.

Cognitive Layer

Having defined FEPs, the communication layer and the physical layer, it is now possible to present the cognitive layer as the third and most sophisticated layer of a collaborative FEP architecture.

The cognitive layer describes the intelligent mechanisms within a human or artificial being that are capable of manipulating, communicating and collaborating intelligently using the defined sensors and effectors provided by the physical layer.

While detailed elaboration of the cognitive layer is beyond the scope of this chapter, we will still briefly touch upon key areas of the cognitive layer: goals, roles and a process for collaboration.

Goals

Understanding the various types of goals that can exist within a collaborative computer game is imperative to understanding the outcomes of the game. Typically, the goals that can be found in such a game are: individual goals, goals of the collective (group) scenario and goals of the play scenario.

In Sect. 3, we have simplified the goal behavior of the play scenario to simply be a single goal defined for the entire play scenario. More complex goal structures within the context of the cognitive layer are beyond the scope of this chapter.

Roles

Roles are the ingredients of a linking mechanism between the cognitive layer and the physical layer. Within the collaborative FEP architecture, roles define specific functions or duties to be performed by FEPs fulfilling the role. A FEP's role can also affect the sensors and effectors available to the being participating in the game.

While our collaborative computer game concept has been designed to operate without the assistance of an overall agent "director", as is the case with work in the field of interactive fiction games (Magerko et al., 2004), we have developed an authority role – The Leader.

The Leader typically is responsible for the organization, initiation and conclusion of a play scenario or defined objective within a collaborative computer game. A Leader may have to organize a team for a single task or may have to organize groups of FEPs over the entire play time of the game.

Depending on how the collaborative computer game has been designed, multiple roles can be defined. In keeping with the *FEP* concept, a human or artificial being is permitted to perform any role defined.

A Collaborative Process

The collaborative process is used to facilitate a formalized process for collaboration during the lifetime of a play scenario. The process draws upon the use of sensors and effectors defined by the physical layer to guide the collaborative processes of a FEP's cognitive layer. The following figure (Fig. 4.3) defines the collaborative process employed within the cognitive layer allowing the participating FEPs to effectively collaborate:

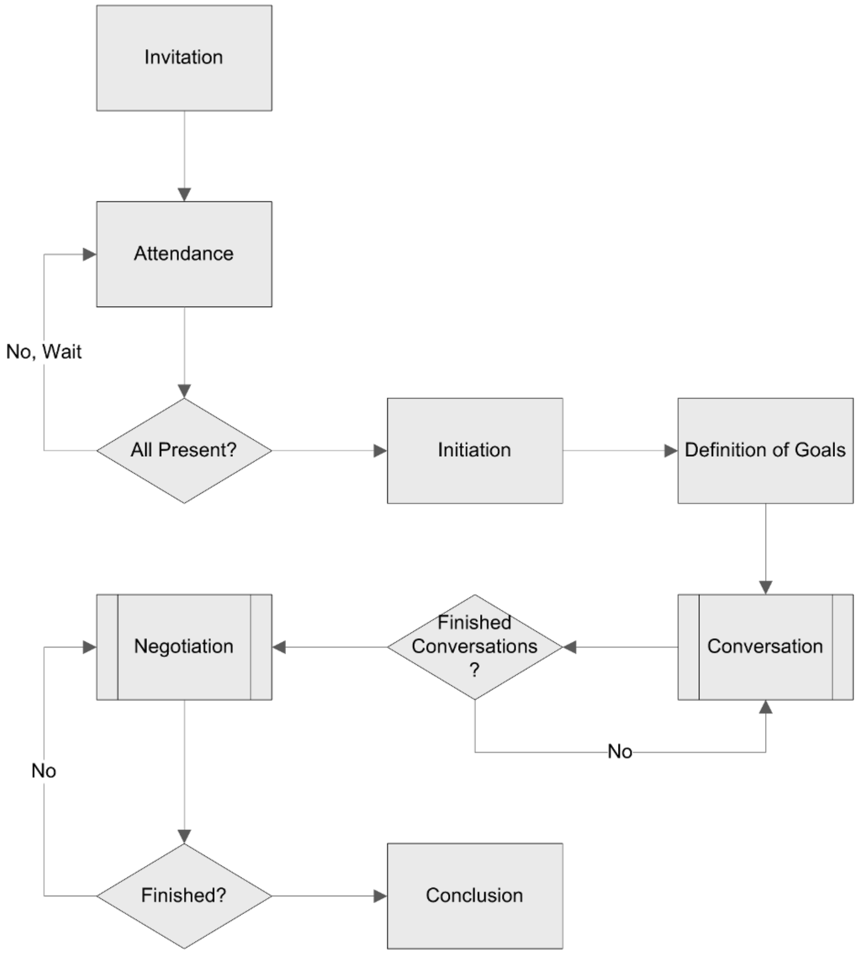


Fig. 4.3. The collaborative process

Invitation

While this phase is not relevant to our current work it bears mentioning that in a collaborative computer game, participation may be by invitation. This process can also include the scheduling of a pre-defined "play time" as well as the roles of the invited FEPs.

Attendance

Once a collaborative computer game has been initiated, participating FEPs are able to "join" the game. Attendance can also occur internally, as human and artificial beings already participating within the game may attend and participate in many play scenarios within the collaborative computer game.

Initiation

At a point determined by the leader (typically when all participants are present, or the scheduled meeting time has been reached) they will declare the play scenario "started". It is at this point that the play scenario can commence. Initiation of the play scenario is actually a special action (we describe actions in more detail later).

Definition of Goals

Before any meaningful collaboration can be achieved between the participating FEPs, it is necessary to define the goals of the current play scenario. This defines the framework for the conversations that will occur during the process. These goals can also be used to determine the success or failure of a particular play scenario (or whether additional play is required).

Presentation

All communication and collaborative behavior within the computer game takes place in the form of "Conversations". Conversations involve the presentation of some instructions, positions, statements, or questions that require additional facts and opinions from the FEPs involved in the collaborative process.

The presentation step may involve physical actions or statements by the partners.

Negotiation

Negotiation involves the willingness of one or more parties involved in the conversation to accept a compromised position. In the collaborative process, this involves the interpretation of the Truth/Facts revealed during the conversation process. As the conversations occur, partners are able to collect truths as well as opinions/positions stated by the other partners. These collected facts or collaborative group knowledge, is then used to feed the negotiation process that attempts to create outcomes based upon the earlier stated goals of the collaborative process.

The negotiation process involves the process of conversation that the partners engage in and allows the FEPs to discover a best fit outcome based upon the goals stated during the definition of goals phase.

Another important factor is how influence plays a role in the interpretation/ negotiation process. The following elements are considered part of the negotiation process.

Questions

Questions are used to obtain truths, facts and perceptions. Questions in a collaborative computer game are any communications made by FEPs that result in an outcome (for simplicity, statements or instructions are also considered "questions"). When an agent proposes a question, there are three possible outcomes: A response (which may be itself another question), an Action or No Response.

Response

A response is given when a FEP receives a directed question, or perceives (through their sensors) the necessity to respond to a question or action. As a response may happen through non-directed communication, but through the perception of other events within the collaborative computer game, a response may itself initiate a new conversation/negotiation. A special type of response that requires the use of effectors not directly related to inter being communication is called an Action.

Actions

Actions are special responses to questions that result in a transition of some item or process from one state to another. For example, if a FEP asked the question "I require a technician for Project X", a possible resulting outcome may be that another participant in the play scenario may perform an action that results in the commencement of a recruitment process to hire a skilled technician for Project X.

Actions tie the collaborative process to the defined physical layer as only those actions available within the physical layer may be enacted to change a defined entity's state. Thus, the introduction of cognitive layer elements results in the ability to enact complex/abstract actions based on perceived physical layer effectors rather than a defined set of actions available for a defined role being enacted by a FEP.

No Response

In some instances, a question may not require a response.

Influence

Collaborative FEPs may create an affinity with one or more entities and are more likely to accept their position during negotiation. Possible methods for obtaining an affinity with one or more FEPs include:

1. The degree to which one FEP's responses convey a perception/opinion that matches that of another FEP. The more that one partner's position matches that of another partner, it becomes more likely that the partner will "trust" the statements of that partner.

2. Some arbitrary influence factor that has the partner tending towards the position of one or more other partners.
3. A pre-existing relationship (for example a friendship) that exists beyond the scope of the collaborative process.

Conclusion

At either a specified time, or when the objectives of the play scenario have been completed successfully, the leader is able to enact a special action that concludes the play scenario.

Prior to the conclusion, the leader or another nominated partner is given the opportunity to summarize or present the outcomes of the scenario to the other participating human and artificial beings. Outcomes can include gauging the success/failure of the play scenario based on the goals defined at the beginning by the leader; can also result on actions required beyond the scope of the current play scenario and could also be the determination that additional play scenarios are required.

Breaking Down the Collaborative Process

Consider a group of FEPs P engaged in the collaborative process c. There will be a set of outcomes O met at the conclusion of the process. The set is based upon the set of defined goals G defined at the beginning of the process and the ability of the partners to collaborate towards the desired outcomes. However there is not a 1:1 ratio of outcomes to goals, and the set of objectives may even be empty.

$$O = c(P, G). \tag{4.1}$$

Each partner p_k is either Human h_i or Artificial a_j. The collaborative group is the union of the human and artificial FEPs.

$$\begin{aligned} P &= \{p_1, .., p_k\} \\ p_k &= \{h_i | a_j\} \\ A &= \{a_1, .., a_j\} \\ H &= \{h_1, .., h_i\} \\ P &= A \cup H \end{aligned} \tag{4.2}$$

During the collaborative process, any partner p_l, where $\mathrm{l} \neq \mathrm{k}$, may ask a question q_m of any other partner p_k in order to receive a response r_m, where $\mathrm{m} = \mathrm{j} + \mathrm{i}$

$$\begin{aligned} r_m = f(p_k, q_m) \text{ where } q_m = g(p_l), \\ r_m = f(p_k, g(p_l)). \end{aligned} \tag{4.3}$$

The response may contain facts or partial knowledge that can be collected and added to the collective knowledge obtained by the group. The collaborative process of the group in order to obtain a set of outcomes is then consensus based upon the interpretation of the group collective knowledge in order to identify whether the partners have achieved (or partially achieved) the initial goals of the group.

The set of *Group Collective Knowledge* K obtained by the group through the collaborative process is a subset of the responses obtained during the collaborative process.

$$\begin{aligned}
&K \subseteq R \\
&K \subseteq \{r_1, .., r_m\} \\
&\{k_1, ..k_q\} \subseteq \{r_1, .., r_m\}
\end{aligned} \tag{4.4}$$

For simplicity, assume that all responses r_m are components of group collective knowledge $K = R$. This means that all results contribute to the set of collective knowledge and that all partners are aware of this knowledge.

$$\begin{aligned}
&K = R \\
&\{k_1, .., k_q\} = \{r_1, .., r_m\} \\
&\quad \text{i.e. } q = m
\end{aligned} \tag{4.5}$$

Outcomes of the collaborative group are a result of the collaborative process between the group of FEPs and the goals of the collaborative process.

$$\begin{aligned}
&O = c(P, G) \\
&O = c(P, G) \\
&O = \{o_1, .., o_n\} \\
&o_n = s(P, n(G, K^P))
\end{aligned} \tag{4.6}$$

where s is a function of all partners P applied to an interpretation function n of the set of goals G, the set of group collective knowledge across the entire set of partners K^P, resulting in an outcome o_n.

4.2.2 A Fuzzy Approach

We have established a formal process by which a collaborative action may take place. What we have not yet discussed is how FEPs within this process are to able make individual intelligent decisions nor at a collaborative decision level.

While it is safe to assume that every human FEP within a collaborative computer game is able to make decisions for himself/herself, in order to create our own play scenario, we need to define how our artificial beings may intelligently assess the information that they receive.

For our purposes, we have selected a fuzzy approach to decision making. The reasons for this decision were its ability to model complex or ill-structured problems, the way in which fuzzy rules can be formulated in an easy to follow IF-THEN manner and its use of human expert knowledge to model the decision making rules.

While more advanced intelligent/learning methodologies could have been used, by selecting fuzzy decision making it is hoped that it will allow a larger audience of multiple skill levels to begin creating FEP systems.

Fuzzy Logic: A Brief Overview

Fuzzy logic is a problem solving concept that enables the use of human heuristic knowledge about a given problem and is capable of solving ill-defined problems. In traditional Boolean logic, answers are either true or false. When dealing with a fuzzy logic, a value may still effectively evaluate to true (1) or false (0), but may also evaluate to any value between the two, giving us "partially true" or "mostly true" values. It is this concept that makes fuzzy logic a useful tool when dealing with complex problems. Fuzzy rules can simplify complex processes by evaluating inputs in order to achieve "best fit" outputs without the need to have exhaustive/complete knowledge of the process. This concept mimics how humans solve problems using heuristic knowledge.

Fuzzy systems encapsulate human expert knowledge of a problem in simplified descriptive rules. The language that is used to describe attributes of a fuzzy system has a certain vagueness to it (hence the use of the term fuzzy) as the language that is used to articulate an attribute's magnitude may apply to more or less of a degree to the attribute being described. Words that are used to describe attributes in a Fuzzy System reflect the way that humans articulate magnitudes. Words such as "cool", "old" and "slow" are used to describe values and are known as *Linguistic Terms*. Just like human experts would describe a value, a linguistic term can describe any input value referred to as a *Crisp Value*, over the universe of discourse. A crisp value however, is more accurately described by some terms than others. This is known as the *Degree of Membership* (DOM) to which a crisp value falls within the range of a linguistic term. For example, Table 4.2 shows a five term linguistic variable for the temperature required to brew coffee:

The DOM to which a crisp value falls within a linguistic term is taken over a numeric range of zero to one.

Table 4.2. Linguistic variable temperature and its terms

Crisp value (temperature, °C)	Linguistic term	DOM
≤87	Cold	0
88	Warm	0.25
93	Brewable	0.5
98	Hot	0.75
≥98	Boiling	1

In order to describe attributes as linguistic terms, the original input values, referred to as *Crisp Values*, are *Fuzzified* and articulated as linguistic variables, for example "temperature", "age" and "speed".

Once fuzzy rules have been applied and there is a result (as a linguistic variable), the fuzzy result must then be *Defuzzified* in order to obtain a crisp output value that can then be applied to the problem.

Take for example a control system that regulates the temperature of an automatic coffee machine. If the ideal temperature of the water being used to brew the coffee needs to maintained at 93°C, then the software needs to measure the temperature of the water within the reservoir and either heat it using an element, or turn off the heating element for a certain amount of time. The temperature sensor takes a reading of 89°C (The crisp value). A set of fuzzy rules can be used to describe this process. These rules are described in an IF–THEN form:

$$\textit{IF temperature} = \textit{warm THEN heating element} = \textit{medium}$$

In the above example, when the temperature of the water is "warm" the heating element will be turned on for a "medium" amount of time. In the above case, the "medium" may equate to sustaining a current to a heating element for 1 min.

The important thing to remember is that when a crisp value is converted into a linguistic term (for example "cold", "warm", "brewable", "hot", "boiling") it will be evaluated based on the *Degree of Membership* (DOM) that it belongs to each term (Table 4.2).

In this chapter we will be using a multiple input single output (MISO) fuzzy system as opposed to more complex multiple input multiple output (MIMO) systems.

The Fuzzification Process

Fuzzification is the process of converting crisp real-world values into linguistic terms. A crisp value may be a member of a number of linguistic terms. The degree of membership that a crisp value has within any one term is determined by the membership function μ_F. This function can take many forms, but result in obtaining a value between 0 and 1 for the crisp value within the universe of discourse.

In the above example (Fig. 4.4), the membership function for considering a temperature "Hot" has resulted in the crisp value having a degree of membership of 0.75. Each linguistic term has its own membership function. When a crisp value is fuzzified, the degree of membership determines the likeliness of the match between the crisp value and the linguistic term.

There are many types of membership functions that are used to describe a linguistic term. The example in shows a Gaussian-shaped membership function. There are many types of membership functions that may be applied

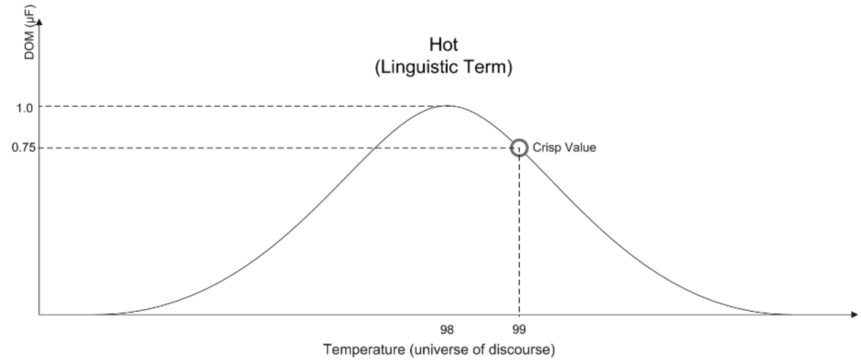


Fig. 4.4. Determining the degree of membership

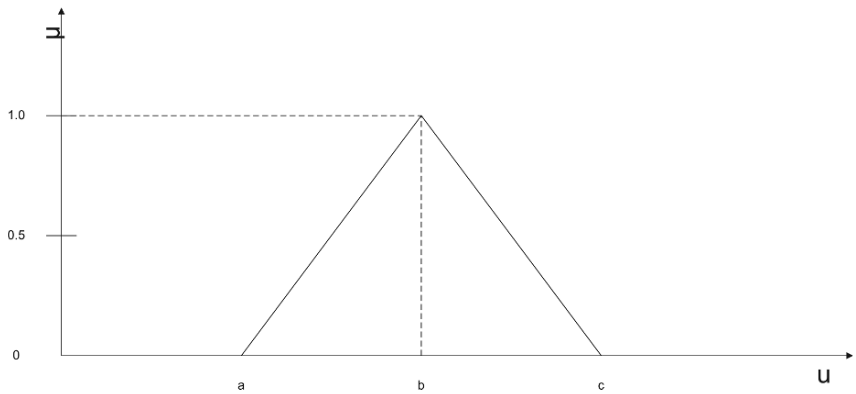


Fig. 4.5. A triangular membership function (T Function)

(they do not even need to be symmetrical); the choice depends upon the application. For the purposes of this chapter and for simplicity, a triangular membership function is used. The triangular membership function (or T-Function) is defined as (Yan et al., 1994):

$$T(u;a,b,c) = \left\{ \begin{matrix} 0 \\ (u-a)/(b-a) \\ (c-u)/(c-b) \\ 0 \end{matrix} \right\} \quad \begin{matrix} \text{For } u < a \\ \text{For } a \leq u \leq b \\ \text{For } b \leq u \leq c \\ \text{For } u > c \end{matrix} \tag{4.7}$$

Where u is an input value from the universe of discourse, while a and c are the lower and upper bounds of the membership function and b is the midpoint (Fig. 4.5).

Fuzzy Rules

As stated earlier, Fuzzy rules are described in terms of IF–THEN conditions. These rules cover all linguistic terms for the required inputs and matches them to conclusions:

$$IF\ x\ is\ A\ THEN\ y\ is\ B$$

As one can imagine, the more linguistic terms there are for a given universe of discourse (crisp input) and the number inputs greatly affects the size of the rule set. In order to determine to what degree a rule applies to the input parameters, a rule's *fire strength* may be calculated. There are many methods that can be used to determine the fire strength of a rule. One method for determining the fire strength of rule is the MAX-MIN.

The MAX-MIN method of determining the fire strength of a particular rule involves taking the degree of membership values for each input into the rule. The fire strength is then determined by the smallest of the fire strengths.

Defuzzification: Obtaining a "Real" output

Once we have achieved an outcome from the application of the fuzzy rules, the resulting fuzzy set values must be converted into a real crisp value. There are a number of methods for selecting an appropriate crisp value including *Center of Gravity, Max Criterion, Mean of Maximum, Center of Area* and *Center-Average.*

In this chapter a centre of gravity (COG) method has been used to determine an appropriate crisp output. The COG method is used in many fuzzy systems given its low computational cost. To obtain a u^{crisp} value we can apply the following to obtain the center of gravity (Passino and Yurkovich, 1998):

$$u^{crisp} = \frac{\sum_i b_i \int \mu_{(i)}}{\sum_i \int \mu_{(i)}}. \tag{4.8}$$

The function $\int \mu_{(i)}$ is used to represent the function required to calculate the area underneath the fuzzy membership function μ_i (where i indicates the ith rule) and b_i is the position where the membership function is at its peak (i.e. has a value of 1). Since it has been indicated that the triangular membership function shall be used in the fuzzy systems involved with the collaborative process, the calculation of the area underneath the triangular membership function becomes (Passino and Yurkovich, 1998):

$$\int \mu_{(i)} = w \left(h - \frac{h^2}{2} \right), \tag{4.9}$$

where w is the width of the triangle's base and h is the fire strength of the fuzzy rule.

Using Fuzzy Logic in a Collaborative System

Coming back to our collaborative process, there are a number of areas where different fuzzy algorithms may be used within a collaborative group of FEPs. The following section shows how an artificial partner would be able to integrate into a collaborative group of FEPs. This would assist us in the understanding of:

- How partners respond during the collaborative process
- How partners interpret group collective knowledge
- How partners obtain outcomes from the collaborative process via negotiation.

FEPs may have differing *perceptions* of the same input values. In order for collaboration to occur effectively, there must be an alignment of perspective. When dealing with a collaborative FEP scenario, it is entirely possible for one partner to refer to something as "large" while another may refer to the exact same source as "small".

The second application of fuzzy logic is in the approximation of one FEP's perspective of scale with their own. As responses in the form of knowledge are articulated to the group of partners, each partner is then able to "align" the response with their own internal perspective.

FEPs participating in the collaborative process are able to approximately align their responses with that of the other partners. It should also be noted that in the responses of the given partners, only one justification has been given for their response. In this chapter, we have constrained the justifications used in the play scenario to one reason. In this case, the justification of a response can be characterized as:

$$\begin{aligned} &r_m \rightarrow j_F \\ &j_F = MAX(MAX(\mu_A(x)), MAX(\mu_B(y))) \end{aligned} \tag{4.10}$$

The response r_m (where r_m is a piece of knowledge) implies a fuzzy justification j_F where j_F in our case is the linguistic term with the highest degree of membership across all inputs.

The resulting fuzzy justification is essentially the conveyance of a linguistic term to other members of the group. This in turn allows the other FEPs to evaluate the responses of other partners in relation to their own.

The justification works on the assumption that while each FEP may have a differing perception for the same inputs, all FEPs articulate their responses in the same linguistic terms (and in the same order). This allows the FEPs to measure the responses of others in relation to their own perception.

For example, if a partner p_k converses with partner p_j using a five term linguistic variable for temperature as defined in Table 4.3 with the crisp value of the temperature being 90°C.

The difference in perception can be simplified to the difference between the linguistic term of one FEP vs. another's perception. In this example, p_j would

Table 4.3. Differing perspectives on the same input

Question		Response and justification	
p_k:	"Turn the coffee brewer on?"	p_j:	"Turn it on for a medium time" "It is warm"
p_j:	"Turn the coffee brewer on?"	p_k:	"Turn it on for a long time" "It is cold"

be able to use the justification of p_k to extrapolate a model of the perception of p_k of the given problem, allowing the FEP to interpret the collective group knowledge supplied by p_k. This then allows p_j to articulate during the negotiation phase of the collaborative process in terms of the perception of inputs by p_k.

Perception does not need be an expensive process in simple scenarios. If all partners articulate their perceptions of the given inputs in the same linguistic terms, the true intention of the FEP is articulated.

The third area within the collaborative process of FEPs that can utilize fuzzy logic is in the negotiation process. At this point, all partners have evaluated the questions and made responses based upon their internal fuzzy reasoning, and all other partners have been able to form a perception of the other partners responses. The negotiation phase of the collaborative process takes the collective group knowledge accumulated during the question process and evaluates the set of outcomes based on the initial goals stated at the beginning of the process. Recall that $o_n = s(P, n(G, K^P))$. A goal g_i must be interpreted against the set of group collective knowledge related to that goal K_i:

$$\begin{aligned} o_n &= s(P, n(G, K^P)) \\ G &= \{g_1, ..g_i\} \\ K^P &= \{{K_1}^P, .., {K_i}^P\} \\ o_n &= s(P, n(\{g_1, ..g_i\}, \{{K_1}^P, .., {K_i}^P\})) \\ o_n &= s(P, \{n(g_1, {K_1}^P), .., n(g_i, {K_i}^P)\}) \end{aligned} \tag{4.11}$$

The interpretation function involves setting a baseline with all group collective knowledge interpreted relative to the baseline. In practice if all partners articulate their perception using the same linguistic terms, this is a trivial operation.

Once the baseline has resulted in a set of Knowledge for the group of FEPs, this set of knowledge can be applied against each goal that the items are related to: $n(g_i, {K_i}^P)$.

In order to satisfy the outcome o_n, the collaborative function s involving all partners and the group collective knowledge interpreted against the baseline is required.

While each FEP will be articulating the group collective knowledge against the baseline, this is not enough to achieve an outcome. Negotiation involves the ability to compromise. In this chapter, we simulate negotiation through the use of an influence factor. This influence factor constraint attracts the resulting partner's decision toward that of another FEP thereby influencing their resulting opinion.

Consider four FEPs that are baristas brewing coffee. The brewing machine has a heating element used to heat water to the right brewing temperature. Using the following linguistic variable to articulate an outcome:

Input: *Temperature* = {*Cold, Warm, Brewable, Hot, Boiling*}
Output: *Make Coffee* = {*Heat, Brew, Heat Off*}

Suppose partner p_1 has had two fuzzy rules that fire based on a temperature input in the form:

IF x IS A_1 *THEN the outcome is* B_1
IF x IS A_2 *THEN the outcome is* B_2

With each rule firing for partner p_1, a final centre of gravity of 3.5 that relates to a linguistic term of *Make Coffee* is achieved. Suppose the partners in Table 4.4 have also evaluated the same rules and determined separate centers of gravity.

During the negotiation phase, we can apply an influence function to change the COG of a given partner's initial fuzzy decision based on the degree of influence the other partners have with the first partner. The influence function that is used in this chapter is simply the sum of the proportion difference between one partner's COG (obtained during the conversation process) and that of another partner:

$$i(COG) = \sum_{1-n, n \neq i} p_{nf}{}^{*}(COG_{pn} - COG_{pi}), \tag{4.12}$$

where $i(COG)$ is the influenced centre of gravity which is the sum of all influence factors multiplied by the difference between the center of gravity of partner p_n and the partner under influence p_i. FEPs using this influence function cannot influence themselves.

The following example shows how the other partners can influence partner p_1's resulting center of gravity. This in turn can potentially change a linguistic term and outcome of the collaborative process (Table 4.5).

Table 4.4. Centers of gravity for each partner

Partner	COG	Linguistic term
p_2	4.4	Heat
p_3	3.7	Brew
p_4	1.2	Heat off

Table 4.5. Influence calculation

		p_2	p_3	p_4
Influence		0.25	0.5	0.1
COG		4.4	3.7	1.2
p_1 value	3.5	0.225	0.1	−0.23
Sum influence factor	0.095			
Initial value + influence factor	3.595			

Once the negotiation phase has been completed, the resulting feedback by all FEPs on the particular outcome can then be evaluated to achieve an outcome. There are many methods for achieving an outcome. In a simple scenario, the outcome can be evaluated by a single partner (normally the leader). In more complex scenarios, a democratic system may be called for requiring the group to reach a majority position.

In the example play scenario, this outcome is achieved by applying a fuzzy decision making approach across the results of the participating FEPs. The final decision, based upon the contributions of the group is performed by the leader.

4.3 Group Decision Making Play Scenario: Software Project Tender Assessment

In the first section of this chapter, we defined what a collaborative FEP is; being either human or artificial in nature, but possessing the capability to collaborate with other FEPs as well as being able to replace any other being, regardless of their underlying nature. In Sect. 2 a formal process for collaborative interaction between human and artificial FEPs was introduced. We discussed the layers of a collaborative architecture, the collaborative process as well as a fuzzy approach to decision making within this process.

In this section, we describe a computer game play scenario where human and artificial beings may collaborate to achieve the collaborative goals of the play scenario.

4.3.1 The Scenario

A large software engineering company is involved in many development projects at any given time. Each project must be judged based on its capability, profitability and risk. The committee that oversees the selection of projects must evaluate each request for tender that the company obtains in order to determine which projects to submit a tender.

In order to determine the most suitable projects, the members of the committee each represent major organisational units within the company. In order for a project to progress to the tender stage, the committee members must find a project that is suitable for all parties.

Since this company is globally dispersed, the members of the committee rarely meet face to face, but rather perform the selection task via an electronic boardroom. In some instances, committee members have been known to use a subordinate to represent their department. In these instances, departments have been known to use an artificial committee member to represent their interests.

The selection committee is overseen by a chairperson who is responsible for managing the meeting, presenting the committee with the various requests for tender and collating the decisions made. In this scenario, the chairperson remains an impartial member of the committee.

All other members of the committee have access to information from their respective departments within the company. Sources of information usually include access to the various systems that manage different areas of the business.

The collaborative process of the electronic meeting room board members will involves six separate roles. Each role represents one of five different organisation units within the company. The additional role is that of the chairperson: the leader role in this scenario. The chairperson is responsible for obtaining tender information from various potential customers and presenting it to the rest of the group for critique. The chairperson is part of the decision making process and is responsible for the successful assessment of tenders during the meeting, however he/she does not express a personal view point on the topics under consideration.

Table 4.6 lists the six participating board members and their role.

David, as the presenter was required to provide the requests for tender to the assessment committee. He provided the tender applications given in Table 4.7 to the committee.

4.3.2 Scenario Collaborative Process

The collaborative process involved in this play scenario involves the assessment of software project tenders for suitability. By using the collaborative process

Table 4.6. FEPs involved in the play scenario

FEP	Role
David	Chairperson
Daniel	Executive
Cathy	Human resources
Ljubo	Project management
Natasha	Finance
Susan	Logistics

Table 4.7. Tender information to be presented to the committee

Customer	Lakeview city council
Tender	Tender for new property rating system
Description	The successful tender shall demonstrate a clear understanding of our Property and Rating requirements as a Local Government Organisation, being able to deliver a new system on time and on budget.
To commence	1/07/2007
Delivery By	1/07/2008
Requirements	
Skill in local govt	Minimum two analysts
Skill in rating	Minimum five analysts
Developers	Approximately 5–12 developers
Delivery	1/07/2008
Tender amount	$250,000
Market segment	Local government
Customer	Australasian Express Courier Services
Tender	Tender for new automated courier tracking system
Description	The successful tender shall provide a system by which shall allow our customers to track their deliveries in real-time via the internet, while managing the transfer, organisation and delivery of these packages.
To commence	1/01/2007
Delivery by	1/07/2007
Requirements	
Developers	Approximately 20–35 developers
Skill in supply chain systems	Minimum two analysts
Web developers	Approximately 7–18 developers
Tender amount	$1,800,000
Market segment	Logistics services
Customer	Western Australia heavy engineering
Tender	Tender for new automated rostering and timecard system
Description	The successful tender shall provide a rostering management system that can integrate with our existing timecard collection devices, as well as provide intelligent rostering for our “Fly In, Fly Out” workforces across numerous mining facilities
To commence	1/07/2007
Delivery by	30/06/2008
Requirements	
Developers	Minimum 10 developers
Business analysts	Approximately three analysts
Skills in roster design	Approximately two analysts
Time and attendance design	Approximately two analysts
Tender amount	1,100,000
Market segment	Mining industry

Table 4.8. A breakdown of the collaborative process

Collaborative step	Play scenario
Invitation	All partners have accepted the invitation to meet and discuss the latest tender requests
Attendance	All partners log into the electronic meeting room
Initiation	Once all partners have entered the electronic meeting room, the presenter starts the meeting Presenter: performs the start meeting action
Definition of goals	The presented states the goals of the electronic meeting Chairperson: "The purpose of the tender projects assessment group is to review incoming requests for tender and determine whether our company should pursue one or more of these tenders"
Presentation	The chairperson presents the collected tenders to the other committee members as defined in the formal conversation procedure detailed in Fig. 4.3
Negotiation	Once a tender has been presented, the negotiation process commences. At this point, the presenter shall establish a baseline for negotiation. Opinions are collected in terms of the baseline terminology
Conclusion	The presenter begins the conclusion phase once all negotiation has been completed. At this point, the presenter provides a summary of the collaborative process. The presenter can then present to the committee the tenders considered the best fit for their organisation. The decision is relayed to the participating partners Finally, the presenter concludes the meeting with the finish action

we can model the process involved in assessing these tenders by the committee members of the electronic meeting room. The following figure highlights how the play scenario is compatible with the collaborative process (Table 4.8).

The conversations required for the tender collaborative process must be defined prior to the construction of the play scenario. Our electronic meeting room will not be utilising "open" (Human-Like) conversations or actions, but for simplicity shall operate within a constrained conversation structure. Figure 4.6 describes this structure, questions and actions that occur every time a new tender is presented to the group of partners.

Once all tenders have been read, the presenter will then announce that Negotiation is to occur. In this process, the committee members must also exchange information amongst themselves, in order to complete their part of the assessment process.

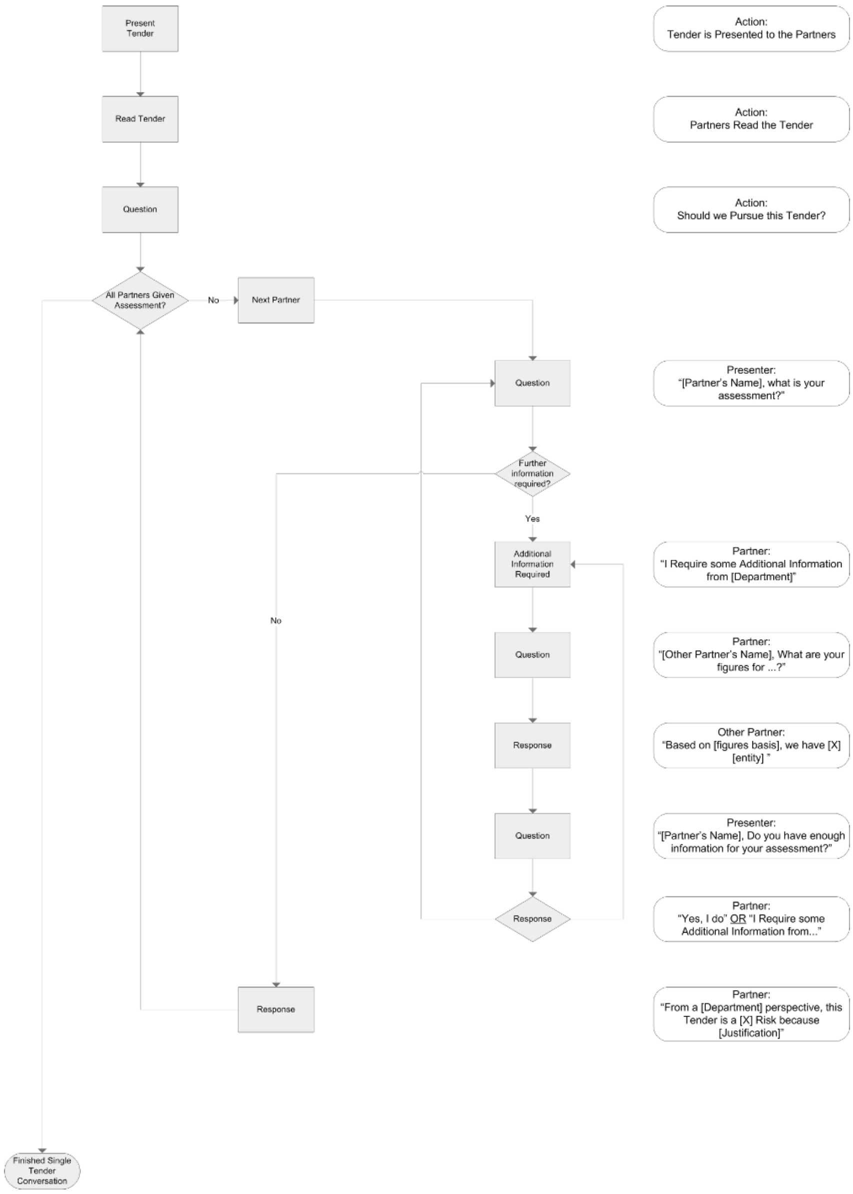


Fig. 4.6. Conversation process structure

4.3.3 Design Assumptions

For this play scenario that we shall be designing, we have made a number of assumptions about the electronic meeting room and the behaviour of the FEPs (the group of committee members) situated within it.

Firstly, the artificial beings do not possess intimate/familiar knowledge of the human that they replace. The intention of the artificial being is to replace a human, giving them the same decision capabilities within the play scenario and not to use the meeting room as a form of Turing Test (A Turing test, made famous by Allan Turing, is a test used to determine if an artificial intelligence is indistinguishable from a human being. A person presents questions to a human and artificial participant that they cannot see, and can only communicate with via a computer terminal. The person presenting questions must then choose which of the interviewees is human).

Secondly, we have limited the number of the decision makers to five FEPs.

Finally, the concept of FEP influence has been reduced to simple factors for the purposes of this the simplicity of presenting this matter in this chapter. Depending on a particular artificial being's affinity with another FEP, it is more likely that that artificial being would choose their position.

4.3.4 Architecture

The architecture of the electronic meeting room is designed around supporting the collaborative process. As such it embodies a layered collaborative architecture. Each layer is interlinked with the next providing a foundation for the collaborative process.

While the layered collaborative architecture is evident across the entire electronic meeting room computer game, the game itself consists of three parts. Firstly, the management of the electronic meeting room is handled by a central meeting room (server) component. The other two parts allow human and artificial FEPs to interact using the meeting room component (Fig. 4.7).

Each of these three parts implements the layered approach. In the design of the electronic meeting room the communication layer handles the transport of information between the electronic meeting room and the FEP interface components. The meeting room, artificial being interface and human being

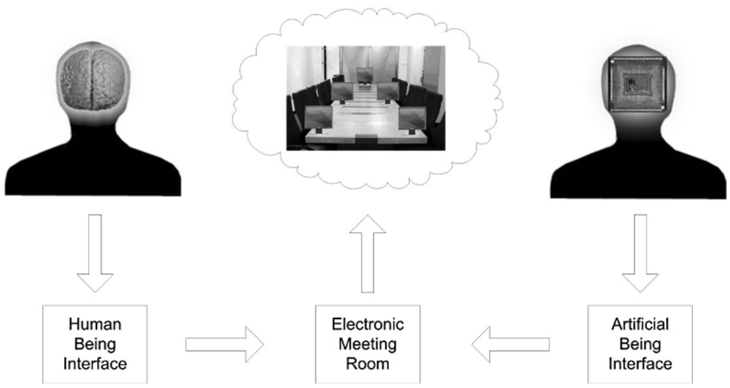


Fig. 4.7. Main software components

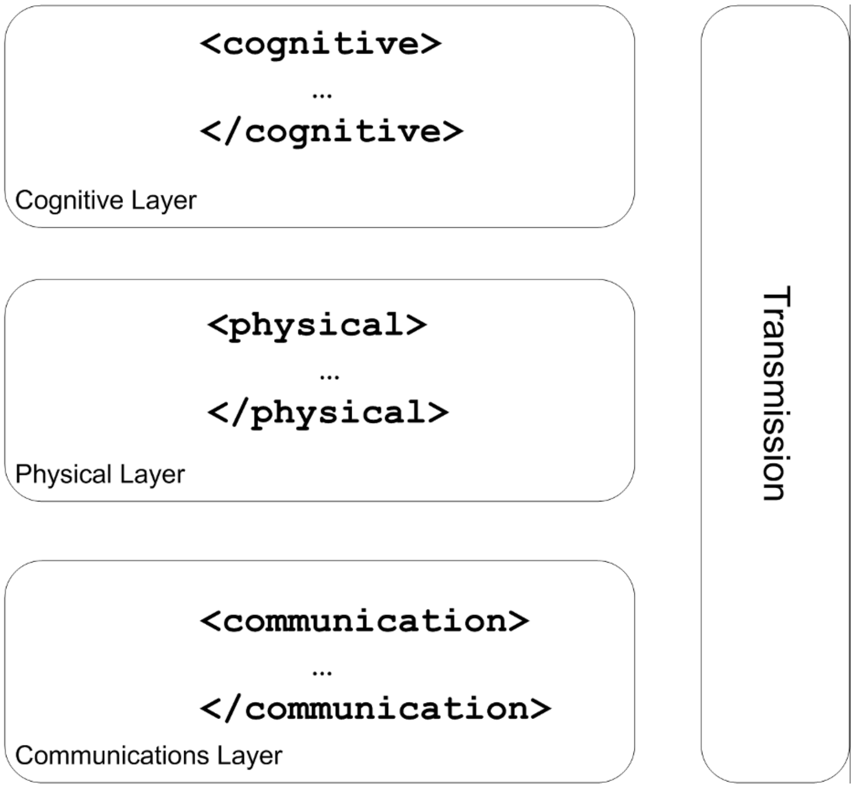


Fig. 4.8. Layered information located in transmission

interfaces operate as independent software components which may exist in a distributed form. Information is transported between each of these components in an XML structure. This allows the discrete separation of information pertaining to each layer (Fig. 4.8).

Each layer is then dealt with in a different manner. Communications layer information is used to handle software-level information such as connections and infrastructure information. The physical layer conveys information about the electronic meeting room and the manipulation of objects within the room (such as a chair, document, etc.). Finally, the cognitive layer conveys messages between each FEP so that the collaborative process may occur (Fig. 4.9).

In the play scenario that we are constructing, we need to define information about the physical layer and what affect it may have upon our tender assessment process.

For the purposes of this play scenario, we have limited the physical layer to four possible collaborative interactions (Table 4.9).

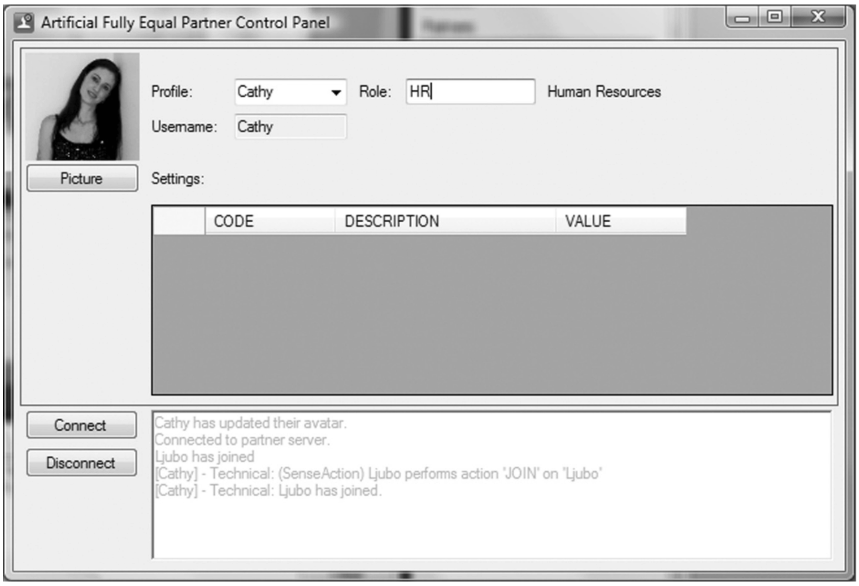


Fig. 4.9. An artificial FEP

Table 4.9. FEP roles and their actions

Action	Description	Role
Begin	Starts the meeting	Chairperson
Finish	Concludes the meeting	Chairperson
Present tender	Presents a request for tender to the other partners	Chairperson
Read tender	Reads/perceives information about a presented tender	Committee member

The cognitive layer is defined in terms of a fuzzy logic based system. For FEPs that are artificial beings, this process is used to determine the resulting decision made during the collaborative process.

4.3.5 Collaborative Process

As stated earlier in this chapter, artificial collaborative FEPs utilise fuzzy logic to determine how to respond during the collaborative process conversations. It is also used to interpret the responses of other partners and to obtain an outcome from the play scenario (the selection of potential projects to pursue based upon the requirements of each organization unit).

Each organization unit has a set of linguistic variables that are used to assess each tender. Some linguistic variables may be obtained via reading each tender's crisp values presented, however other linguistic variables require

Table 4.10. Input linguistic variables and terms for the play scenario

Role	Inputs	Outputs
Executive	Profit market segment	Corporate risk
Financial	Revenue	Financial risk
	Expenditure	
Human resources	Skills required	HR/recruitment risk
	Skill availability	
Project management	Lead time	Project management risk
	Project duration	
Logistics	Equipment outlay	Logistical risk

Table 4.11. Output linguistic variable and terms

Risk analysis	Crisp output value
Low risk	1
Medium risk	3
High risk	5

the committee member to ask another member to obtain crisp data. Each role has its own set of linguistic variables (Table 4.10).

The output decision of each partner is articulated using the various risk linguistic variables. These risk variables are then used to determine the final assessment result for each tender presented. Each of the risk variables has three terms: low, medium and high risk (based upon the perspectives of each committee member). The combined fuzzy rule table amounts to 243 assessment rules (Table 4.11).

In the next section, we discuss how the electronic meeting room's underpinning software operates within the given scenario.

4.3.6 The Software

Each human FEP that has joined the electronic meeting room perceives the room via the human interface component. They see visually themselves placed at the meeting room table along with the other partners. By default, partners only see a silhouette of all other board members in the meeting room however an image/avatar may also be nominated.

Figure 4.10 shows how Daniel has joined the meeting room. Each of the other committee members is also present and all members have nominated an image/avatar to represent themselves in the meeting room. During this meeting, Daniel is unaware whether his fellow committee members are human or artificial beings (Fig. 4.9).

From this point on, we shall discuss how the artificial FEPs behaved whilst within the play scenario. To demonstrate this, we conducted the entire play scenario with artificial FEPs.

Fig. 4.10. Human FEP software component

The artificial partners involved in the scenario used different fuzzy rules. In order to reduce the complexity of the rule set, the number of output linguistic terms presented by each FEP was reduced to three. This leads to a final fuzzy rule assessment table consisting of 243 rules. Given that each artificial partner may have a different perspective when applying the fuzzy rules to each of the submitted tenders, it can be quickly seen that there is a significant number of rules to be designed and evaluated.

The following table is a small sample of the fuzzy rules that were constructed from heuristic information provided by a number of human experts (Table 4.12).

In order to evaluate the inputs in terms of the fuzzy rules stated above, we have utilised the T-Function to determine the degree of membership each crisp value has to each of the three linguistic terms of each input. To achieve this, each membership function required values for the variables defined in (4.7). The following values were used:

1. Center b point: defined as being the value indicated by each partner for each linguistic term. (for example: Cathy defined low recruitment as being less than 5% and hence is her centre point the low recruitment membership function).
2. To determine the a and c values, we defined a "bandwidth" value for each linguistic variable. The a and c values are equal to $b - 1/2$ *bandwidth value* and $b + 1/2$ *bandwidth value*, respectively.

Table 4.12. Table of play scenario fuzzy rules

RULE	IF	HR Risk =	Projects Risk =	Financial Risk =	Logistical Risk =	Executive Risk =	Then	Suitability =
11	IF	LOW	LOW	MED	LOW	MED	Then	HIGHSUIT
12	IF	LOW	LOW	MED	LOW	HIGH	Then	HIGHSUIT
13	IF	LOW	LOW	MED	MED	LOW	Then	HIGHSUIT
14	IF	LOW	LOW	MED	MED	MED	Then	SUITABLE
15	IF	LOW	LOW	MED	MED	HIGH	Then	SUITABLE
16	IF	LOW	LOW	MED	HIGH	LOW	Then	HIGHSUIT
17	IF	LOW	LOW	MED	HIGH	MED	Then	SUITABLE
18	IF	LOW	LOW	MED	HIGH	HIGH	Then	UNSUITABLE
19	IF	LOW	LOW	HIGH	LOW	LOW	Then	HIGHSUIT
20	IF	LOW	LOW	HIGH	LOW	MED	Then	HIGHSUIT
21	IF	LOW	LOW	HIGH	LOW	HIGH	Then	HIGHSUIT
22	IF	LOW	LOW	HIGH	MED	LOW	Then	HIGHSUIT
23	IF	LOW	LOW	HIGH	MED	MED	Then	SUITABLE
24	IF	LOW	LOW	HIGH	MED	HIGH	Then	UNSUITABLE
25	IF	LOW	LOW	HIGH	HIGH	LOW	Then	HIGHSUIT
26	IF	LOW	LOW	HIGH	HIGH	MED	Then	UNSUITABLE
27	IF	LOW	LOW	HIGH	HIGH	HIGH	Then	UNSUITABLE
28	IF	LOW	MED	LOW	LOW	LOW	Then	HIGHSUIT
29	IF	LOW	MED	LOW	LOW	MED	Then	HIGHSUIT
30	IF	LOW	MED	LOW	LOW	HIGH	Then	HIGHSUIT

The following sample demonstrates part of the collaborative process in action:

David `Should we pursue this tender?`
(Action) `A new tender is being presented`

(Presenting Tender for New Automated Rostering and Timecard System)

(Action) `Cathy performs action ‘READ’ on ‘TENDER’`
(Action) `Ljubo performs action ‘READ’ on ‘TENDER’`
David `Cathy, what is your assessment?`
Cathy `I require additional information from Project Management`
Cathy `Ljubo, what are your figures for Available Skills?`
Ljubo `Based on Available Skills, we have 49 Percent Availability`
David `Cathy, Do you have enough information for your assessment?`
Cathy `Yes I Do`
David `Cathy, what is your assessment?`
Cathy `From a Human Resources perspective, this Tender is a Low Risk because of Percentage Recruitment Required and Available Skills`

Table 4.13. Linguistic variables utilised by human resources artificial partner

Linguistic variable	linguistic term	Crisp value
Skill availability (%)	Low	25
	Medium	50
Bandwidth = 50	High	75
Recruitment	Low	5
Required (%)	Medium	10
Bandwidth = 10	High	15

Table 4.14. Fuzzy table used by the human resources artificial partner

IF	Skill availability =	AND	Recruitment Required =	THEN	HR Risk =
IF	HIGH	AND	LOW	THEN	LOW
IF	HIGH	AND	MED	THEN	LOW
IF	HIGH	AND	HIGH	THEN	HIGH
IF	MED	AND	LOW	THEN	MED
IF	MED	AND	MED	THEN	MED
IF	MED	AND	HIGH	THEN	HIGH
IF	LOW	AND	LOW	THEN	HIGH
IF	LOW	AND	MED	THEN	HIGH
IF	LOW	AND	HIGH	THEN	HIGH

When Cathy is assessing the merits of this particular tender, as an artificial being, the information in Table 4.13 was used.

The artificial partner representing the Human Resources department of this company applies its assessment of the tender based upon availability of resources required (i.e. Employees that will be available to work on this project) as well as any recruitment effort required to offset any shortfall in skills. To achieve this, the artificial partner requires information from the Tender, as well as their Project Management counterpart. Once this information is collected, the following table of fuzzy rules is applied to achieve an assessment in terms of human resource requirements (Table 4.14).

The resulting application of the equations specified in (4.7) resulted in the following results being recorded and then used by the artificial partner to respond with their assessment of the tender:

(Cathy - Technical): `Fuzzy Evaluation Called`

```
BEGIN - DoFuzzyProcess()
    Result Sum Area: 4.8456
    Result Sum BPointArea: 9.252
    Centre of Gravity: 1.909


   FINISH - DoFuzzyProcess() result:
```
HRRISK is LOW

Table 4.15. Conclusion phase stating the final outcomes

Conclusions given	Centre of gravity (Equation (4.8), as applied to rules shown by Table 4.14)
Based on your contributions, the tender "Tender for New Property Rating System" is SUITABLE for our business to pursue	2.9365
Based on your contributions, the tender "Tender for New Automated Courier Tracking System" is UNSUITABLE for our business to pursue	1.8148
Based on your contributions, the tender "Tender for New Automated Rostering and Timecard System' is SUITABLE for our business to pursue	2.9365

Conclusion

The final part of the collaborative process is the conclusion phase. In the electronic meeting room, the final decision based on the feedback provided by all partners is to be made by the Chairperson. The following process is used to determine a final outcome of the electronic meeting room:

1. An average COG is determined for each tender.
2. Since the outcome is to obtain a tender within the suitable to highly suitable range of assessments, the averages for each tender are then evaluated using the membership functions for suitable and highly suitable linguistic terms. Recall that the b Point for Suitable and Highly Suitable are 3 and 5 respectively with a bandwidth of 3.
3. The resulting centre of gravity across the average and good linguistic terms determines the final outcome value.
4. The final assessment of each tender as determined by the group are as follows:

By following through the collaborative processes within the electronic meeting room play scenario, we have been able to ascertain that only two of the presented tenders were suitable for our business to pursue.

This scenario demonstrates the layered approach to design and development of computer game-based collaborative decision-making play scenarios. Figure 4.11 shows how the tender evaluation system consists of three distinct layers. The communications layer, implemented using technologies such as XML and service-based communication. The physical layer defining the objects and actions available to the committee members, and finally the cognitive layer, consisting of the cognitive process as well as interfaces that permit the communication between human and artificial beings situated within the tender evaluation system.

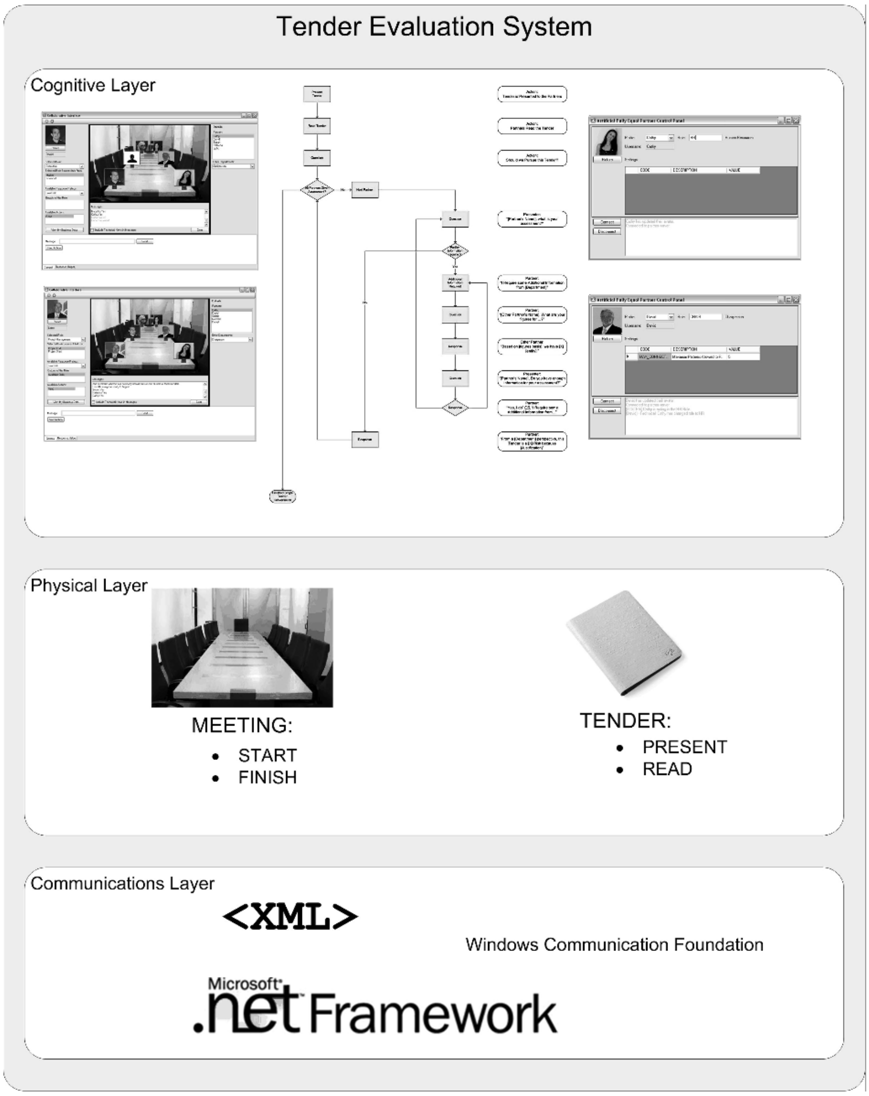


Fig. 4.11. Layered architecture of the tender evaluation system

4.4 Concluding Remarks

In this chapter we have explored the concept of human and artificial beings as FEPs in collaborative decision making situations. Collaborative FEPs are fully replaceable or interchangeable with any other FEP and are not necessarily aware that the other partners are human or artificial beings.

We have presented a method for dealing with collaboration between FEPs by using a structured collaborative process. Within this process, we have shown how fuzzy logic can be applied to achieve collaborative outcomes and have demonstrated the application of these concepts through a simple play scenario.

This collaborative scenario also demonstrates that interaction amongst human and artificial beings as FEPs is possible in many fields, which broadens the application of this application for potential use in social interactions.

4.4.1 Additional Considerations

As discussed earlier, for simplicity the electronic meeting room play scenario had its scope constrained in a number of areas. These areas provide opportunity to consider additional improvements to the play scenario. Architecturally, the system maintains fuzzy rules, scenarios, roles and settings via a generic database structure, allowing it to be easily extended in this fashion.

Firstly, the play scenario can be extended to a support a greater number of participants within the collaborative process.

The number of inputs into the play scenario may be extended to consider additional areas of consideration for the FEPs.

Changes can be made to the decision-making processes being used within the artificial FEPs to include memory in the fuzzy decision making process. It would also be possible to completely replace the fuzzy decision making components with a different intelligent decision-making methodology that would be used within the collaborative process.

There are also many architectural design features (both implicit and explicit) that support FEPs in collaborative computer games. Some of these desirable features that support a computer game as a collaborative FEP element were identified by Thomas and Vlacic (2003). The result of this work was determining key software design attributes necessary to facilitate effective collaboration within computer games. These attributes are an effective guide when designing collaborative computer games from an architectural perspective.

While a collaborative computer game is a vehicle for intelligent, cognitive game play, it is also our research platform. As such, it requires certain attributes and interfaces necessary for the study of collaborative beings. The following architectural properties support a cooperative game platform that facilitates collaborative FEPs play scenarios:

Exogenous events. (Hanks et al., 1993) in order to emulate the adaptive, collaborative and cognitive abilities of real world (embodied) beings within a collaborative computer game, we must introduce into play scenarios an element of unexpected change to the state of play (or, as is the case with experimentation, *manufacture* these unplanned events if required).

Causal structure. A complex causal structure is necessary to imitate the complex cause and effect actions and reactions of real-world scenarios. Our approach provides a causality structure necessary to exhibit complex collaborative (and cognitive) abilities in the FEPs situated within the game. Causality is realized through rules defined by the physical layer, as well as more complex rules determined by the cognitive layer.

A concept of time. The collaborative computer game and the FEPs situated within it must be able to operate within a linear time environment. For the purposes of the game, a play scenario defined by the physical layer may have a very simple time structure (based on a sequence of events and/or triggers) or a more complex real-time system where effective collaboration may require the ability to respond within a finite time span.

Support for experimentation. having the ability to control the conditions within the game, thus allowing for repeatable, quantifiable play scenarios (Vincent et al., 2000).

A well-defined interface between the collaborative computer game and beings that are situated within it is necessary to support true autonomy of the FEPs within the game and encourages collaborative behavior.

In addition to these desirable features that support a collaborative computer game were also a number of practical considerations identified when selecting or constructing a collaborative computer game. While not directly related to supporting a layered architecture approach, practical features will affect the embodiment of a collaborative computer game:

The availability and cost of infrastructure and development tools required;

The learning curve required in order to be completely familiar with the underlying infrastructure used to construct human and artificial software interfaces;

Environmental complexity was also identified as an important factor in creating an effective collaborative computer game. This becomes more of an issue for the scalability of complex elements (especially the causal structure), as we move from simpler play scenarios to the more complex, introducing more sophisticated elements to the cognitive layer.

Documentation (or lack thereof) is a strong factor for and against a particular tool or infrastructure. When selecting the necessary tools to construct a collaborative computer game, availability of adequate reference material and support structure is imperative so as not to detract from constructing an effective realization of the concept with distracting technical issues.

4.4.2 Other Applications for Collaborative Decision Making

In this chapter we have discussed collaborative FEPs in the context of computer games, demonstrating such a system in action by way of the electronic meeting room play scenario. While the concept of collaborative FEPs is compelling in today's software industry, where intelligent artificial players interact and collaborate effectively with human players, there are many other fields of endeavour that can benefit from this concept.

As businesses look for more ways to gain an edge over their competitors, training and recruitment form a major part of the work done by Human Resource departments in large businesses. The cost of training alone for a large organisation can be staggering. We see collaborative FEPs being used as a tool in self guided training, group training and recruitment. An organisation may develop a training course that is taught by an artificial FEP, or group learning activities where humans and artificial beings collaborate to practice teamwork, acting in particular roles where one or more humans may require training. It could also be used as a tool in the recruitment process to gauge a recruit's responses when confronted with a team-based scenario.

While we have concentrated on a linguistic form of communication in our exploration of the collaborative process, there are many non verbal methods of conveying ideas and intentions to other FEPs.

One of the biggest challenges in research and development is intelligent automated transportation. One of the challenges facing researchers in this field is how intelligent transport systems can operate *within* our current system rather than being in its own separate/contained transport network.

By using a collaborative FEP approach, it is possible to integrate human drivers and intelligent automated transport systems are FEPs participating in the collaborative process of moving from one place to another efficiently and safely. Artificial beings in this category of transport system would be able to communicate to each other, while sensors are able to detect the intentions of human drivers (such as a turning indicator) allowing artificial drivers to collaborate with the other vehicles on the road.

Socially-oriented computer games are another area where collaborative human and artificial beings acting as FEPs find potential application. This application may be of potential use in the fields of clinical psychology and behavioural studies. There are many open questions about how application of this technology may be used as a "safe" environment to assist those in need of specialised social/behavioural assistance. It is upon us, as collaborative beings to investigate with our colleagues in these disciplines and examine potential application in this field as collaborative fully equal partners.

References

Beinisch Y, Paunov C, Vickery G, Vincent, SW (2005): Digital Broadband Content: The online computer and video game industry Organisation for Economic Cooperation and Development Report (OECD)

Hanks S, Pollack M, Cohen P (1993): Benchmarks, Testbeds, Controlled Experimentation, and the Design of Agent Architectures AI Magazine, AAAI Press, 14:17–42

Jennings NR, Wooldridge M (1995): Intelligent Agents: Theory and Practice Knowledge Engineering Review, 10

Laird JE (2001) Using a Computer Game to Develop Advanced AI IEEE Computer, 34:70–75

Magerko B, Laird JE, Assanie M, Kerfoot A, Stokes D (2004): AI characters and directors for interactive computer games, Proceedings of the 2004 Innovative Applications of Artificial Intelligence Conference

Passino KM, Yurkovich S (1998): Fuzzy Control, Addison-Wesley, Reading, MA

Riedl M, Saretto CJ, Young MR (2003): Managing interaction between users and agents in a multiagent storytelling environment, Proceedings of the Second International Conference on Autonomous Agents and Multi-Agent Systems

Thomas D, Vlacic L (2003): Selecting an environment for cooperative autonomous robot research. In: Hyungsuck Cho JK, Lee J (ed.), Intelligent Robots: Vision, Learning and Interaction. KAIST Press, pp. 187–198

Thomas D, Vlacic L (2005): TeamMATE: Computer Game Environment for Collaborative and Social Interaction IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO'05)

Vincent R, Horling B, Lesser V (2000): Experiences in Simulating Multi-Agent Systems Using TAEMS, The Fourth International Conference on MultiAgent Systems (ICMAS 2000), AAAI

Yan J, Ryan M, Power J (1994): Using Fuzzy Logic, Prentice-Hall, Englewood Cliffs, NJ

5

Decision Analysis with Fuzzy Targets

Van-Nam Huynh[1], Yoshiteru Nakamori[1], and Tetsuya Murai[2]

[1] Japan Advanced Institute of Science and Technology, 1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan, {huynh, nakamori}@jaist.ac.jp
[2] Hokkaido University, Kita-ku, Sapporo 060-0814, Japan, murahiko@main.ist.hokudai.ac.jp

Summary. It is widely accepted that a common precept for the choice under uncertainty is to use the expected utility maximization principle, which was established axiomatically. Recently, a formal equivalence between this principle of choice and the target-based principle, that suggests that one should select an action which maximizes the (expected) probability of meeting a (probabilistic) uncertain target, has been established and extensively discussed. This chapter introduces the fuzzy target-based model for a class of decision making under uncertainty problems. Two methods for inducing utility functions from fuzzy targets are introduced, along with an interestingly established link between the decision maker's different attitudes about target and different risk attitudes in terms of utility functions. In addition, we also introduce how the fuzzy target-based approach can provide a unified way for solving the problem of fuzzy decision making with uncertainty about the state of nature and imprecision about payoffs.

5.1 Introduction

A classical problem in decision analysis under uncertainty is to rank a set of alternatives defined on a state space S, where, due to the uncertainty in the state of nature, each alternative a may lead to different outcomes taking from a set of outcomes D, usually associated with a random outcome $X_a : S \to D$. The decision maker (DM) must then use some ranking procedure over alternatives for making decisions. The most commonly-used ranking procedure is based on the expected utility model. The DM defines a utility function U over D and then ranks an alternative a by its expected utility $EU(X_a)$. Note that the utility function U is bounded and unique up to increasing affine transformations (or *cardinal*, for short) (Savage 1954).

Another ranking procedure is that the DM establishes some *target* t and then ranks an alternative a by the probability $P(X_a \succeq T)$ that it meets the target (Manski 1988). Although simple and appealing from this target-based point of view, the DM may not know for sure which target he should select. Then he could define some random variable T as his uncertain target (or,

V.-N. Huynh et al.: *Decision Analysis with Fuzzy Targets*, Studies in Computational Intelligence (SCI) **97**, 135–155 (2008)
www.springerlink.com

benchmark) instead and rank an alternative a by the probability $P(X_a \succeq T)$ that it meets the uncertain target T (or, it outperforms the benchmark), provided that the target T is stochastically independent of the random outcomes to be evaluated. We call this procedure *target-based* or *benchmarking*.

Interestingly, these two different procedures are shown to be both mathematically and observationally equivalent (LiCalzi 1999). In particular, Castagnoli and LiCalzi (1996) discussed a formal equivalence of von Neumann and Morgenstern's expected utility model (Von Neumann and Morgenstern 1944) and the target-based model with reference to preferences over lotteries. Later, a similar result for Savage's expected utility model (Savage 1954) with reference to preferences over acts was established by Bordley and LiCalzi (2000). Despite the differences in approach and interpretation, both target-based procedure and utility-based procedure essentially lead to only one basic model for decision making. It should be worth, however, emphasizing that while both target-based and utility-based decision making demand an understanding of probabilities, the utility-based model additionally requires a comprehension of cardinal utilities. More details on the formal connection between the utility-based approach and the target-based approach in decision analysis with uncertainty can be referred to, e.g., (Abbas and Matheson 2005; Bordley 2004; Castagnoli and LiCalzi 2006; LiCalzi 1999).

As discussed above, while the target-based decision model satisfies the Savage axioms (Savage 1954) serving as an axiomatic foundation for rational decision making under uncertainty, it also maintains the appealing features from the target-based approach as thinking about targets is very natural in many practical situations of decision making. In addition, it is also natural to think of the target-based decision model using fuzzy targets instead of random ones, because in many contexts where due to a lack of information, defining fuzzy targets is much easier and intuitively natural than directly defining random targets. This chapter aims at introducing a fuzzy target-based approach to decision analysis with uncertainty, which has been recently studied in (Huynh et al. 2006a, b; Huynh et al. in press).

Firstly, we will introduce a fuzzy target-based model for the classical problem of decision making under uncertainty (DUU, for short), in which different representations of a fuzzy target may result in different utility functions; where utility can be interpreted as probability of meeting the fuzzy target. From this target-oriented point of view, DM be able to express *attitudes about the target selection*, and then an interesting link between different attitudes about target and different risk attitudes in terms of utility functions can be established. It would be worth noting here that recently Yager (Yager 1999, 2000) has focused on the construction of decision functions allowing for the inclusion of information about decision attitude and probabilistic information about the uncertainty. However, Yager's valuation-based approach does not consider the risk attitude factor in terms of utility functions as in the traditional utility-based paradigm, while focusing on a mechanism for combining probabilistic information about state of nature with information about DM's attitude in the formulation of a valuation function.

Then, we will discuss the issue of how this target-based approach can be applied for fuzzy decision analysis under uncertainty. Interestingly, it will be seen that the fuzzy target-based approach provides an appealing and unified one for fuzzy decision making with uncertainty. Note that in the fuzzy set based method (Rommelfanger 2004), we may first apply Zadeh's extension principle (Nguyen 1978) to obtain the fuzzy expected utility for each alternative and then utilize either a defuzzification method or a ranking procedure for fuzzy numbers for the purpose of making the decision. Consequently, different results may be produced if different methods of ranking fuzzy numbers or defuzzification are used. However, this difference in results does not clearly reflect the influence of the DM's attitude. In addition, a bunch of methods for ranking fuzzy numbers developed in the literature (e.g., (Bortolan and Degani 1985; Chen and Hwang 1992)) also makes it even difficult for DM in choosing a most appropriate method for each particular problem.

The rest of this chapter is organized as follows. Section 2 introduces some basic notions of fuzzy sets and their representations, which will be used in the subsequent sections to transform fuzzy targets so as to allow the application of the target-based decision model for a class of decision making under uncertainty problems. Section 3 briefly presents a target-based interpretation of the expected utility model. Section 4 explores a target based decision model using fuzzy targets, in which different attitudes of the DM on target are also discussed in relation to the concept of risk attitude. Section 5 then briefly describes how the fuzzy target-based approach could be possibly extended for applying to fuzzy decision analysis. Finally, some concluding remarks and future work are presented in Sect. 5.6.

5.2 Fuzzy Sets and Their Representations

Let U be a universe of discourse. A fuzzy set F of U is a mapping from U into the unit interval: $\mu_F : \mathsf{U} \rightarrow [0,1]$, where $\mu_F(x)$ is called the membership degree of x in F. For $\alpha \in (0,1]$, the α-cut F_α of is a crisp subset of U defined as

$$F_\alpha = \{x \in \mathsf{U} | \mu_{\mathrm{F}}(x) \geq \alpha\}$$

In fuzzy set theory, the concept of α-cuts plays an important role in establishing the relationship between fuzzy sets and crisp sets. Intuitively, each α-cut F_α of a fuzzy set F can be viewed as a crisp approximation of F at the level $\alpha \in (0,1]$.

In the case where a fuzzy set F has a discrete membership function, i.e.,

$$F = \{(x_k, \mu_{\mathrm{F}}(x_k))\}, \text{ for } x_k \in \mathsf{U} \text{ and } k = 1, \ldots, N$$

with N being a finite positive integer, Dubois and Prade (Dubois and Prade 1987) pointed out that the family of its α-cuts forms a nested family of focal elements in terms of Dempster-Shafer theory (Shafer 1976). In particular, assuming the range of the membership function μ_F, denoted by $\text{rng}(\mu_F)$, is $\text{rng}(\mu_F) = \{\alpha_1, \ldots, \alpha_n\}$, where $\alpha_i > \alpha_{i+1} > 0$, for $i = 1, \ldots, n-1$, then the so-called *body of evidence* induced from F is defined as the collection of pairs

$$\mathcal{F}_F = \{(F_{\alpha_i}, \alpha_i - \alpha_{i+1}) | i = 1, \ldots, n\}$$

with $\alpha_{n+1} = 0$ by convention. Then the membership function μ_F can be expressed by

$$\mu_F(x_k) = \sum_{x_k \in F_{\alpha_i}} m_i \tag{5.1}$$

where $m_i = \alpha_i - \alpha_{i+1}$ that can be viewed as the probability that F_{α_i} stands as a crisp representative of the fuzzy set F (Dubois and Prade 1989), and so F_F is referred to as a consonant random set. Note that the normalization assumption of F insures the body of evidence does not contain the empty set.

A fuzzy number A is defined as a fuzzy set with the membership function $\mu_A(x)$ of the set of R all real numbers that satisfies the following properties (Zimmermann 1985):

- A is a normal fuzzy set, i.e., $\sup_{x \in \mathsf{R}} \mu_A(x) = 1$;
- A is a convex fuzzy set, i.e. $\mu_A(\lambda x_1 + (1-\lambda)x_2) \geq \min(\mu_A(x_1),\ \mu_A(x_2))$ for $\forall x_1,\ x_2 \in \mathsf{R}$ and $\lambda \in [0,1]$;
- the support of A, i.e. the set $\text{supp}(A) = \{x \in \mathsf{R} \mid \mu_A(x) > 0\}$, is bounded.

According to (Klir 2006), a fuzzy number A can be conveniently represented by the canonical form

$$\mu_A(x) = \begin{cases} f_A(x), & a \leq x \leq b \\ 1, & b \leq x \leq c, \\ g_A(x), & c \leq x \leq d, \\ 0, & \text{otherwise} \end{cases}$$

where $f_A(x)$ is a real-valued function that is monotonically increasing, and $g_A(x)$ is a real-valued function that is monotonically decreasing. In addition, as in most applications, we assume that functions f_A and g_A are continuous. If $f_A(x)$ and $g_A(x)$ are linear functions then A is called a trapezoidal fuzzy number and denoted by $[a,b,c,d]$. In particular, $[a,b,c,d]$ becomes a triangular fuzzy number if $b = c$.

For any fuzzy number A expressed in the canonical form, its α-cuts are expressed for all $\alpha \in (0,1]$ by the formula (Klir 2006)

$$A_\alpha = \begin{cases} [f_A^{-1}(\alpha), g_A^{-1}(\alpha)], & \text{when } \alpha \in (0,1), \\ [b,c], & \text{when } \alpha = 1. \end{cases} \tag{5.2}$$

where f_A^{-1} and g_A^{-1} are the inverse functions of f_A and g_A, respectively. In the case that A degenerates into a crisp interval, i.e., $A = [a,b]$, we define $A_\alpha = A$ for all $\alpha \in (0,1]$.

In the case of a fuzzy number A that possesses a continuous membership function, as discussed in Dubois and Prade (1989), the family $\{A_\alpha \mid \alpha \in (0,1]\}$ can be viewed as a uniformly distributed random set, consisting of the Lebesgue probability measure on [0,1] and the set-valued mapping $\alpha \mapsto A_\alpha$. Then the membership function μ_A is expressed as an integral:

$$A_\alpha = \int_0^1 \mu_{A_\alpha}(x) d\alpha \tag{5.3}$$

where μ_{A_α} is the characteristic function of crisp set A_α.

In computer applications, a fuzzy number A can be usually approximated by sampling the membership function along the membership axis. That is, assuming uniform sampling and that the sample values are taken at membership grades $\alpha_1 = 1 > \alpha_2 > \ldots > \alpha_{n-1} > \alpha_n > 0$, then, from the perspective of the above interpretation of fuzzy sets, we can approximately represent A as

$$F_A = \{(A_{\alpha_i}, \alpha_i - \alpha_{i+1}) | i = 1, \ldots, n\} \tag{5.4}$$

and then membership degrees can be approximately computed via (1), the discrete version of (3). The approximation becomes better when the sample of membership grades is finer. This approximate representation of fuzzy numbers has been either implicitly or explicitly used in literature by many authors, particularly in the issue of ranking fuzzy numbers, e.g., (Chen and Lu 2001; Fortemps and Roubens 1996; Saade 1996).

5.3 Target-Based Model of the Expected Value

In this chapter we consider the problem of decision making in the face of uncertainty that can be most effectively described using the decision matrix shown in Table 5.1 (see, e.g., (Brachinger and Monney 2004; Yager 1999)). In this matrix, A_i $(i = 1, \ldots, n)$ represent the alternatives available to a DM, one of which must be selected. The elements S_j $(j = 1, \ldots, m)$ correspond to the possible values/states associated with the so-called state of nature S. Each element c_{ij} of the matrix is the payoff the DM receives if the alternative A_i

Table 5.1. Decision matrix

Alternatives	State of nature			
	S_1	S_2	$\ldots$	S_m
A_1	c_{11}	c_{12}	$\ldots$	c_{1m}
A_2	c_{21}	c_{22}	$\ldots$	c_{2m}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
A_n	c_{n1}	c_{n2}	$\ldots$	c_{nm}

is selected and state S_j occurs. The uncertainty associated with this problem is generally a result of the fact that the value of S is unknown before the DM must choose an alternative A_i. Let us consider the decision problem as described in Table 5.1 with assuming a probability distribution P_S over $S = \{S_1, \ldots, S_m\}$. Here, we restrict ourselves to a bounded domain of the payoff variable that $D = [c_{\min},\ c_{\max}]$.

As is well-known, the most commonly used method for valuating alternatives A_i to solve the DUU problem described by Table 5.1 is to use the expected utility value:

$$v(A_i), \triangleq EU_i = \sum_{j=1}^{m} P_S(S_j)U(c_{ij}) \tag{5.5}$$

where U is a utility function defined over D.

On the other hand, each alternative A_i can be formally considered as a random payoff having the probability distribution P_i defined, with an abuse of notation, as follows:

$$P_i(A_i = x) = P_S(\{S_j : c_{ij} = x\}) \tag{5.6}$$

Then, the target-based model (Bordley and LiCalzi 2000) suggests using the following value function

$$\begin{aligned} v(A_i) &= P(A_i \succeq T) \\ &= \sum_{x} P(x \succeq T)P_j(A_i = x) \\ &= \sum_{j=1}^{m} P_S(S_i)P(c_{ij} \succeq T) \end{aligned} \tag{5.7}$$

where the random target T is stochastically independent of any random payoffs A_i, and $P(x \succeq T)$ is the cumulative distribution function (c.d.f., for short) of the target T.

Recall that the utility function U is bounded and increasing. Thus, after having normalized its range to the unit interval [0,1], U has all the properties of a cumulative distribution function over the payoff domain D. As shown in (Bordley and LiCalzi 2000), by a standard probability-theoretic argument, one can associate to the c.d.f. U a random payoff T stochastically independent of A_i and then view $U(x)$ as the probability that x meets the target T, i.e., $U(x) = P(x \succeq T)$. This makes (5) and (7) formally identical. In other words, the target-based decision model with decision function $v(A_i)$ in (7) above is equivalent to the expected utility model defined by (5).

In the next section, we will introduce two assessment procedures for $U(x)$ in the case that the DM can only assess his target in terms of a possibility distribution instead.

5.4 A Decision Model Based on Fuzzy Targets

Before discussing about the problem of decision making using fuzzy targets, it is necessary to recall that when expressing the value of a variable as a fuzzy set, we are inducing a possibility distribution (Zadeh 1978) over the domain of the variable. Formally, the soft constraint imposed on a variable V in the statement "V is F", where F is a fuzzy set, can be considered as inducing a possibility distribution Π on the domain of V such that $\mu_F(x) = \Pi(x)$, for each x. Here, by a fuzzy target we mean a possibility variable T over the payoff domain D represented by a possibility distribution $\mu_T : D \to [0,1]$. For simplicity, we assume further that T is normal, convex and has a piecewise continuous function with $\text{supp}(T) = [c_{\min}, c_{\max}]$, where $\text{supp}(T)$ denotes the support of T.

Let us turn back to the DUU problem described in Table 5.1. In a target-based decision model, assume now that the DM establishes a fuzzy target T which reflects his attitude. Then, according to the optimizing principle, after assessed the target the DM would select an act as the best that maximizes the expected probability of meeting the target defined by

$$v(A_i) = \sum_{j=1}^{m} P_S(S_j)\mathsf{P}(c_{ij} \succeq T) \tag{5.8}$$

where $\mathsf{P}(c_{ij} \succeq T)$ is a formal notation indicating the *probability of meeting the target* of value c_{ij} or, equivalently, the utility $U(c_{ij}) \triangleq \mathsf{P}(c_{ij} \succeq T)$ in the utility-based language.

5.4.1 Normalization-Based Method

A direct and simple way to define $\mathsf{P}(c_{ij} \succeq T)$ is making use of Yager's method (Yager 1999, 2002) for converting a possibility distribution into an associated probability distribution via the simple normalization. Particularly, the possibility distribution μ_T of the target T is first converted into its associated probability distribution, denoted by P_T, as follows

$$P_T(t) = \frac{\mu_T(t)}{\int_{c_{\min}}^{c_{\max}} \mu_T(t)dt}$$

Then $\mathsf{P}(c_{ij} \succeq T)$ is defined as the c.d.f as usual by

$$\mathsf{P}(c_{ij} \succeq T) \triangleq U_1^T(c_{ij}) = \int_{c_{\min}}^{c_{ij}} P_T(t)dt \tag{5.9}$$

It should be noted that this definition of $\mathsf{P}(c_{ij} \succeq T)$ is also formally used, but without a probabilistic interpretation, for the so-called *satisfaction function* $S(T < c_{ij})$ in (Lee-Kwang and Lee 1999) for the comparison between a fuzzy number T with a crisp number c_{ij}. A formulation of DUU using fuzzy targets based on this approach has been studied in (Huynh et al. 2006a).

5.4.2 Alpha-Cut Based Method

Now we introduce another method for inducing the utility function associated with $\mathsf{P}(c_{ij} \succeq T)$ based on the α-cut representation of fuzzy target T.

According to the probabilistic interpretation of (4), we can now approximate $\mathsf{P}(x \succeq T)$ by

$$\mathsf{P}(x \succeq T) \cong \Delta\alpha \sum_{i=1}^{k} P(x \succeq T_{\alpha_i}) \tag{5.10}$$

Where $\Delta\alpha$ is the separation between any two adjacent levels, and

$$P(x \succeq T_{\alpha_i}) = \begin{cases} 0, & \text{if } x \leq t_l(\alpha_i) \\ \dfrac{x - t_l(\alpha_i)}{t_r(\alpha_i) - t_l(\alpha_i)}, & \text{if } t_l(\alpha_i) \leq x \leq t_r(\alpha_i) \\ 1, & \text{if } x \geq t_r(\alpha_i) \end{cases} \tag{5.11}$$

i.e., the c.d.f. of the random variable having a uniform distribution over T_{α_i} that is viewed as an approximation of T at level α_i.

Clearly, the right side of the expression (10) is the Riemann sum of the function $f(\alpha) = P(x \succeq T_\alpha)$ over [0,1] with respect to the partition $\alpha_1, \ldots, \alpha_{k+1}$. Thus, we generally define $\mathsf{P}(x \succeq T)$ as

$$\mathsf{P}(x \succeq T) = \int_0^1 P(x \succeq T_\alpha) d\alpha \tag{5.12}$$

The approximation in (10) of the integral in (12) improves the finer the sample of membership grades.

Returning to our DUU problem described as above we obtain the following utility function induced from the fuzzy target T:

$$\mathsf{P}(c_{ij} \succeq T) \triangleq U_2^T(c_{ij}) = \int_0^1 P(c_{ij} \succeq T_\alpha) d\alpha \tag{5.13}$$

5.4.3 Fuzzy Targets and Risk Attitude

Let us now consider four fuzzy targets which correspond to prototypical attitudes of DM on target assessment. The first one expresses a *neutral* behavior of the DM on target and is represented by the possibility distribution $T_{\text{neutral}}(x) = 1$ for $c_{\min} \leq x \leq c_{\max}$, and $T_{\text{neutral}}(x) = 0$, otherwise. Then, it is easily shown that both methods for inducing utility yield the same value function for (8):

$$v(A_{\mathrm{i}}) = \sum_{j=1}^{m} \frac{c_{\mathrm{ij}} - c_{\min}}{c_{\max} - c_{\min}} P_{\mathrm{S}}(S_{\mathrm{j}})$$

which is equivalent to the expected value model.

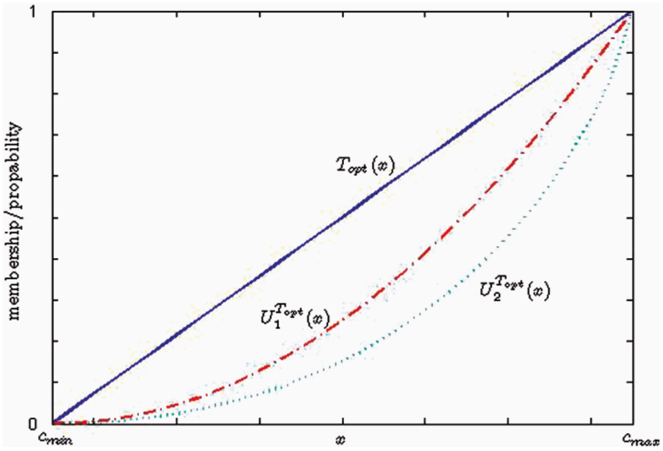


Fig. 5.1. Optimistic target

The second is called the *optimistic target*. This target would be set by a DM who has an aspiration towards the maximal payoff. Formally, the optimistic fuzzy target, denoted by T_{opt}, is defined as follows

$$T_{opt}(x) = \begin{cases} \dfrac{x - c_{\min}}{c_{\max} - c_{\min}}, & \text{if } c_{\min} \leq x \leq c_{\max} \\ 0, & \text{otherwise} \end{cases}$$

Figure 5.1 graphically depicts the membership function $T_{opt}(x)$ along with the utility functions $U_1^{T_{opt}}(x)$ and $U_2^{T_{opt}}(x)$ corresponding to this target.

The third target is called the *pessimistic target*. This target is characterized by a DM who believes bad things may happen and has a conservative assessment of the target, which corresponds to ascribing high possibility to the uncertain target being a low payoff. The membership function of this target is defined by

$$T_{pess}(x) = \begin{cases} \dfrac{c_{\max} - x}{c_{\max} - c_{\min}}, & \text{if } c_{\min} \leq x \leq c_{\max} \\ 0, & \text{otherwise} \end{cases}$$

The portraits of related functions corresponding to the pessimistic target are shown in Fig. 5.2.

Consider now the fourth target linguistically represented as "*about* c_0" whose membership function is defined by

$$T_{\tilde{c}_0}(x) = \begin{cases} \dfrac{x - c_{\min}}{c_0 - c_{\min}}, & c_{\min} \leq x \leq c_0 \\ \dfrac{c_{\max} - x}{c_{\max} - c_0}, & c_0 \leq x \leq c_{\max} \\ 0, & \text{otherwise} \end{cases}$$

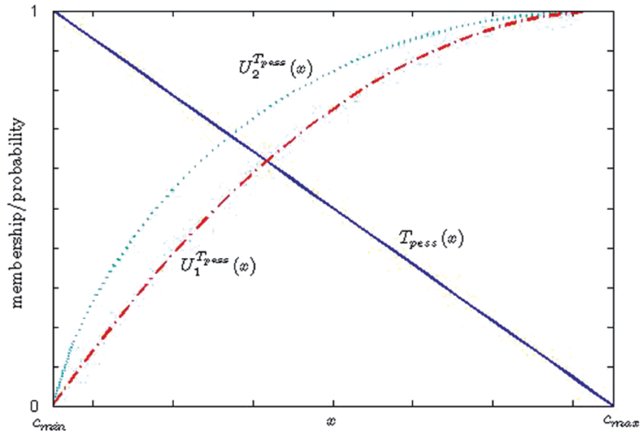


Fig. 5.2. Pessimistic target

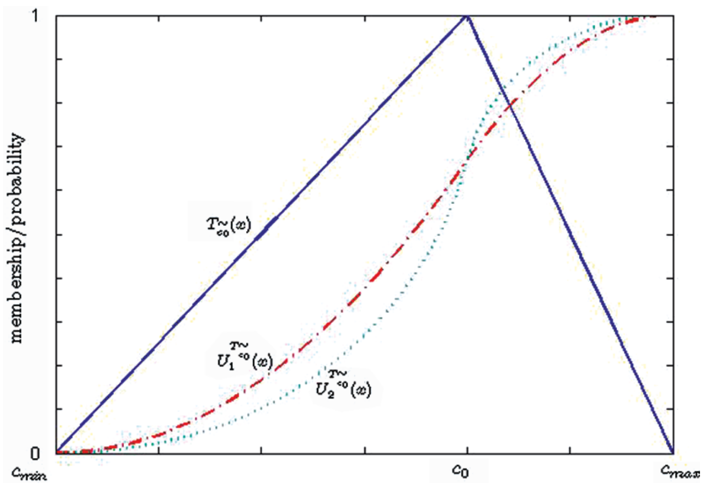


Fig. 5.3. Unimodal target

where $c_{\min} \leq c_0 \leq c_{\max}$. This fuzzy target characterizes the situation at which the DM establishes a modal value c_0 as the most likely target and assesses the possibilistic uncertain target as distributed around it. We call this target the *unimodal*. Figure 5.3 graphically illustrates for this situation.

Looking at Figs. 5.1–5.3, we see that the portraits of the utility functions $U_1^T(x)$ and $U_2^T(x)$ have similar shapes for each corresponding target. However, the behavior of the utility function $U_1^T(x)$ is steeper towards the modal value of the corresponding targets than that of the utility function $U_2^T(x)$. This practically implies that the value function $v(\cdot)$ defined with utility function $U_2^T(x)$ reflects a stronger decision attitude towards the target than that defined with utility function $U_1^T(x)$ as shown in the example below.

As we have seen from Fig. 5.1, the optimistic target T_{opt} leads to the convex utility functions and therefore, exhibits a risk-seeking behavior. This is because of having an aspiration towards the maximal payoff; the DM always feels loss over the whole domain except the maximum, which would produce more risk-seeking behavior globally. By contrast, Fig. 5.2 shows that the pessimistic target induces the concave utility functions and thus equivalently corresponds to global risk-aversion behavior. More interestingly, the unimodal target induces the utility functions that are equivalent to the S-shape utility function of Kahneman and Tversky's prospect theory (Kahneman and Tversky 1979), according to which people tend to be risk averse over gains and risk seeking over losses. In the fuzzy target-based language, as the DM assesses his uncertain target as distributed around the modal value, he feels loss (respectively, gain) over payoff values that are coded as negative (respectively, positive) changes with respect to the modal value. This would lead to the behavior consistent with that described in the prospect theory. A link of this behavior to unimodal probabilistic targets has been established by LiCalzi in (LiCalzi 1999). Further, it has been also suggested in the literature that this sort of target be the most natural one to occur.

5.4.4 An Illustration Example

Let us consider the following simple example taken from (Brachinger and Monney 2004) to illustrate the point discussed above.

Connie is the owner of a bakery and every early Sunday morning she has to prepare some cakes that will hopefully be sold during the day. Due to cakes being made with a special kind of cream, the unsold cakes must be thrown away at the end of the day. The selling price of a cake is \$15.25 and the production cost of a cake is \$5.75. Though Connie does not know how many cakes will be purchased by customers on a particular Sunday, by experience she assumes that the demand will not exceed five cakes. If she wants to have a chance of making any profit, she should prepare a few cakes. On the other hand, if she prepares too many cakes, it may happen that there will not be enough customers to buy all of them. The question is how many cakes should she prepare?

From the verbal description as above, we now formalize the decision problem by means of a decision matrix D as follows. Let x denote the number of cakes Connie is going to prepare. So there are six possible values of x ranging between 0 and 5, called alternatives. Let y denote the total number of cakes requested by the customers on a particular Sunday. Clearly, y is also an integer between 0 and 5 and the value of y s is a matter of chance. Each possible value of y corresponds to a state of nature. Then we have a decision matrix D of dimension 6×6, where each element $d_{ij} (i, j = 1, \ldots, 6)$ corresponds to Connie's profit if she decides to make $i - 1$ cakes and the actual demand of cakes is $j - 1$. It is easy to see that

$$d_{ij} = \begin{cases} 15.25j - 5.75i - 9.5, & \text{if } i \geq j \\ 9.5(i-1), & \text{if } i < j \end{cases}$$

and then the decision matrix D is shown in Table 5.2.

This decision problem clearly is a decision problem under uncertainty because the consequence of choosing any number of cakes to prepare depends on the unknown total number of cakes requested by the customers. Now, assume further that, based on some experience on the consumption behavior of clients, Connie is able to specify a probability distribution on the states of nature as follows: $P(y = 0) = P(y = 4) = P(y = 5) = 0.1,\ P(y = 1) = P(y = 3) = 0.2,\ P(y = 2) = 0.3$.

With this decision problem, we can define the profit domain D as the interval $[-28.75, 47.5]$. Then, Table 5.3 shows the computational results of two value functions with different fuzzy targets for alternatives, where

$$v_1(A_i) = \sum_{j=1}^{m} p_j U_1^T(c_{ij}) \text{ and } v_2(A_i) = \sum_{j=1}^{m} p_j U_2^T(c_{ij})$$

and Table 5.4 shows the corresponding ranking results for the decision of number of cakes to prepare.

From the ranking result shown in Table 5.4, we see that both value functions $v_1(\cdot)$ and $v_2(\cdot)$ suggest the same solution for the selection problem in the

Table 5.2. Connie's profit matrix

Alternatives (x)	States (y)					
	0	1	2	3	4	5
$A_1(x = 0)$	0.00	0.00	0.00	0.00	0.00	0.00
$A_2(x = 1)$	−5.75	9.50	9.50	9.50	9.50	9.50
$A_3(x = 2)$	−11.50	3.75	19.00	19.00	19.00	19.00
$A_4(x = 3)$	−17.25	−2.00	13.25	28.50	28.50	28.50
$A_5(x = 4)$	−23.00	−7.75	7.50	22.75	38.00	38.00
$A_6(x = 5)$	−28.75	−13.50	1.75	17.00	32.25	47.50

Table 5.3. The target-based value functions

Value functions	Targets	Alternatives					
		A_1	A_2	A_3	A_4	A_5	A_6
$v_1(\cdot)$	Neutral	0.3770	0.4816	0.5462	0.5508	0.5154	0.4600
	Optimist	0.1422	0.2356	0.3160	0.3434	0.3280	0.2920
	Pessimist	0.6119	0.7277	0.7765	0.7582	0.7028	0.6280
$v_2(\cdot)$	Neutral	0.3770	0.4816	0.5462	0.5508	0.5154	0.4600
	Optimist	0.0820	0.1440	0.2049	0.2342	0.2347	0.2266
	Pessimist	0.7451	0.8295	0.8579	0.8377	0.7885	0.7138

Table 5.4. The ranking result

Value functions	Targets	Ranking
$v_1(\cdot)$	Neutral	$A_4 \succ A_3 \succ A_5 \succ A_2 \succ A_6 \succ A_1$
	Optimist	$A_4 \succ A_5 \succ A_3 \succ A_6 \succ A_2 \succ A_1$
	Pessimist	$A_3 \succ A_4 \succ A_2 \succ A_5 \succ A_6 \succ A_1$
$v_2(\cdot)$	Neutral	$A_4 \succ A_3 \succ A_5 \succ A_2 \succ A_6 \succ A_1$
	Optimist	$A_5 \succ A_4 \succ A_6 \succ A_3 \succ A_2 \succ A_1$
	Pessimist	$A_3 \succ A_4 \succ A_2 \succ A_5 \succ A_1 \succ A_6$

cases that the DM has a neutral (equivalently, who abides by the expected value) or a pessimistic behavior in assessing the target. However, in the case of an optimistic-oriented DM, A_4 is ranked at first with the value function $v_1(\cdot)$, while A_5 becomes the first with the value function $v_2(\cdot)$. This shows that the target-based decision model using $U_2^T(\cdot)$ reflects a stronger decision attitude towards the target than that using $U_1^T(\cdot)$. More generally, this can be intuitively observed by the fact that the spread of the difference of the value function $v_2(\cdot)$ between opposite-oriented targets is much larger than that of the value function $v_1(\cdot)$. Also note that the computational results of these two functions are different except, obviously, for the case of the neutral target.

As we have seen, depending on the fuzzy target assessed, different courses of action will be selected with different expected probabilities of meeting the target produced. Practically, if the decision maker was experiencing with an initial loss or series of losses in the past, she would preclude continuation of loss and then the pessimistic target would be considered as an appropriate one. However, if the decision maker has sufficient capital to absorb potential losses, she may then assess the neutral or even the optimistic target. It would also be of interest to note that the nature of the target assessment may be also influenced, to some extent, by the personal philosophy of the decision maker. That is, a cautious decision maker may have a pessimistic-oriented target, while a more adventurous entrepreneur may prefer an optimistic-oriented target.

5.5 Fuzzy Decision Analysis

5.5.1 Target-Based Decision Procedure

As having discussed above, the fuzzy target-based method of uncertain decision making is formally equivalent to a procedure which, once having designed a target T, consists of the following two steps:

1. For each alternative A_i and state S_j, we define

$$p_{ij} = \mathsf{P}(c_{ij} \succeq T) \tag{5.14}$$

Table 5.5. The derived decision matrix

Alternatives	State of nature			
	S_1	S_2	$\dots$	S_m
A_1	p_{11}	p_{12}	$\dots$	p_{1m}
A_2	p_{21}	p_{22}	$\dots$	p_{2m}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
A_n	p_{n1}	p_{n2}	$\dots$	p_{nm}

and then form a "probability of meeting the target" table described in Table 5.5 from the payoff table (i.e., Table 5.1).

2. Define the value function as the expected probability of meeting the target

$$v(A_i) = \sum_{j=1}^{m} p_j p_{ij} \tag{5.15}$$

We now consider the problem of decision making under uncertainty where payoffs may be given imprecisely. Let us turn back to the general decision matrix shown in Table 5.1, where c_{ij} can be a crisp number, an interval value or a fuzzy number. Clearly in this case we have an inhomogeneous decision matrix and traditional methods can not be applied directly. One of methods to deal with this decision problem is to use fuzzy set based techniques with help of the extension principle and many procedures of ranking fuzzy numbers developed in the literature. In the following we introduce a fuzzy target-based procedure for solving this problem, for more details, see (Huynh et al. 2006; Huynh et al. in press).

5.5.2 Normalization-Based Model

Firstly using the preceding mechanism, once having assessed a fuzzy target T, we need to transform the payoff table into the one of probabilities of meeting the target. For each alternative A_i and state S_j, the probability of payoff value c_{ij} meeting the target, also denoted by $p_{ij} = \mathsf{P}(c_{ij} \succeq T)$, is defined as in the following.

If c_{ij} is a crisp number, as previously discussed (refer to (9)) we have

$$p_{ij} = \frac{\int_{-\infty}^{c_{ij}} T(x)dx}{\int_{-\infty}^{+\infty} T(x)dx} \tag{5.16}$$

If c_{ij} is an interval value or a fuzzy number, the procedure for computing p_{ij} is as follows.

In the case where c_{ij} is an interval value, say $c_{ij} = [a, b]$, we consider c_{ij} as a random variable with the uniform distribution on $[a, b]$. If c_{ij} is a fuzzy quantity represented by a possibility distribution F_{ij}, we have the associated probability distribution of F_{ij} defined by

$$P_{F_{ij}}(x) = \frac{F_{ij}(x)}{\int_{-\infty}^{+\infty} F_{ij}(x)dx}$$

and also denote, with an abuse of notation, c_{ij} as the random variable associated with the distribution $P_{F_{ij}}$. Recall that the associated probability distribution of the target T is

$$P_T(x) = \frac{T(x)}{\int_{-\infty}^{+\infty} T(x)dx}$$

and we also use the same notation T for the random variable associated with the distribution P_T.

Having considered c_{ij} and T as two random variables in both these cases with an acceptance of the independent assumption of c_{ij} and T, we can define the probability of c_{ij} meeting the target T as

$$\begin{aligned} p_{ij} &= \mathsf{P}(c_{ij} \succeq T) \\ &= \int_{-\infty}^{+\infty} P_T(x)P(c_{ij} \succeq x)dx \end{aligned} \tag{5.17}$$

where similar as in (9) we have

$$P(c_{ij} \succeq x) = \int_{x}^{+\infty} P_{F_{ij}}(y)dy$$

It should be also emphasized here that in the research topic of ranking fuzzy numbers, the authors in (Lee-Kwang and Lee 1999) proposed a ranking procedure based on the so-called satisfaction function (SF, for short), which is denoted by S and defined as follows. Given two fuzzy numbers A and B,

$$S(A > B) = \frac{\int_{-\infty}^{+\infty} \int_{y}^{+\infty} \mu_A(x) \circ \mu_B(y)dxdy}{\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \mu_A(x) \circ \mu_B(y)dxdy} \tag{5.18}$$

where $\circ$ is a T-norm and S $(A > B)$ is interpreted as "the possibility that A is greater than B" (or, the evaluation of A in the local viewpoint of B). Clearly, by a simple transformation we easily show that the probability p_{ij} of c_{ij} meeting the target T defined in (17) above is the SF $S(c_{ij} > T)$ with T-norm selected as being the multiplication operator.

5.5.3 Alpha-Cut Based Model

In the preceding section, we have introduced a method of inducing probabilities of meeting the target $p_{ij} = \mathsf{P}(c_{ij} \succeq T)$ based on the membership function representation of fuzzy numbers and simple normalization (Yager et al. 2001). Here, we will introduce another method based on the alpha-cut representation of fuzzy numbers and a probability-based comparison relation over intervals (Huynh et al. in press).

Particularly, in the case that c_{ij} is an interval value or a fuzzy number, we define

$$p_{ij} = \mathsf{P}(c_{ij} \succeq T) = \int_0^1 P(c_{ij_\alpha} \succeq T_\alpha) d\alpha \tag{5.19}$$

where

$$P(c_{ij_\alpha} \succeq T_\alpha) = \int_{-\infty}^{+\infty} f_{c_{ij}^\alpha}(x) \left[\int_{-\infty}^{+\infty} f_{T_\alpha}(y) dy \right] dx \tag{5.20}$$

and $f_{c_{ij}^\alpha}(x)$ and $f_{T_\alpha}(y)$ are uniform distributions over intervals c_{ij}^α and T_α, respectively. Note that in the case that c_{ij} is an interval value, as previously defined, we have $c_{ij}^\alpha = c_{ij}$ for all $\alpha \in (0, 1]$. Further, in the case that both intervals c_{ij}^α and T_α degenerate into scalar numbers, we define by convention

$$P(c_{ij_\alpha} \succeq T_\alpha) = \begin{cases} 1, \text{if } c_{ij_\alpha} > T_\alpha, \\ \dfrac{1}{2}, \text{if } c_{ij_\alpha} = T_\alpha, \\ 0, \text{if } c_{ij_\alpha} < T_\alpha. \end{cases} \tag{5.21}$$

More details on this method as well as interpretations and motivations for it could be referred to (Huynh et al., in press).

As such, we have transformed the (possibly, inhomogeneous) fuzzy decision matrix into the derived decision matrix described by Table 5.5 (Abbas and Matheson 2005), where each element p_{ij} of the derived decision matrix can be uniformly interpreted as the probability of payoff c_{ij} meeting the target T. From this derived decision matrix, we can then use the value function (15) for ranking alternatives and making decisions. It is worth emphasizing here that as an important characteristic of this target-based approach to fuzzy decision analysis, it also allows for including the DM's attitude, which is expressed in assessing his target, into the formulation of decision functions. Consequently, different attitudes about target may lead to different results of the selection.

5.5.4 A Numerical Example

For illustration, let us consider the following application example adapted from (Rommelfanger 2004). LuxElectro is a manufacturer of electro-utensils and currently the market demand for its products is higher than the output. Therefore, the management is confronted with the problem of making a decision on possible expansion of the production capacity. Possible alternatives for the selection are as following:

- A_1: Enlargement of the actual manufacturing establishment with an increase in capacity of 25%.

- A_2: Construction of a new plant with an increase in total capacity of 50%.
- A_3: Construction of a new plant with an increase in total capacity of 100%.
- A_4: Renunciation of an enlargement of the capacity, the status quo.

The profit earned with the different alternatives depends upon the demand, which is not known with certainty. Due to the amount of information the management estimates three states of nature corresponding to "high", "average" and "low" demand with associated prior probabilities of 0.3, 0.5 and 0.2, respectively. Then the prior matrix of fuzzy profits $\tilde{U}_{ij}$ (measured in million Euro) is given in Table 5.6, where fuzzy profits are represented parametrically by triangular and trapezoidal fuzzy numbers.

Using the extension principle in fuzzy set theory, we obtain the expected profits of alternatives as shown in Table 5.7 below, where risk neutrality is assumed. Then to make a decision one can apply one of ranking methods developed in the literature on these fuzzy profits. Intuitively, one can see that the alternatives A_4 and A_1 are much worse than the alternatives A_3 and A_2. However, it is not so easy to say which alternative is dominated by the other among these two better alternatives. However, if using for example the centroid of fuzzy numbers as the ranking criterion we get the ranking order as $A_2 \succ A_3 \succ A_1 \succ A_4$.

To apply the target-based procedure suggested above for solving this problem, according to the information given by this problem, we define the domain of profits as $D = [-90, 230]$. Then, using the above procedures for inducing probabilities p_{ij} and (15), we obtain the computational results for the value function with different fuzzy targets as shown in Table 5.8. Table 5.9 shows the ranking orders of alternatives corresponding to different targets and methods.

From Table 5.9 we see that the result of both models reflect very well the behavior of the DM which is expressed in assessing the target. In particular,

Table 5.6. Fuzzy profit matrix $\tilde{U}_{ij} = \tilde{U}(A_i, S_j)$

Alternatives	States		
	$S_1 : 0.3$	$S_2 : 0.5$	$S_3 : 0.2$
A_1	(80;90;100;110)	(75;85;90;100)	(50;60;70)
A_2	(135;145;150;165)	(120;130;140)	$(-40; -30; -20)$
A_3	(170;190;210;230)	(100;110;125)	$(-90; -80; -70; -60)$
A_4	70	70	70

Table 5.7. Expected fuzzy profits via extension principle

Alternatives	Expected fuzzy profit	Centroid value
A_1	(71.5;81.5;87;97)	84.25
A_2	(92.5;102.5;104;115.5)	103.73
A_3	(83;96;104;119.5)	100.76
A_4	70	70

Table 5.8. Results of the value function

Methods	Targets	Alternatives			
		A_1	A_2	A_3	A_4
Normalization	*Neutral*	0.5445	*0.6051*	0.5957	0.50
	Optimist	0.2983	0.411	*0.4461*	0.25
	Pessimist	0.7907	*0.7997*	0.7466	0.75
Alpha-Cut	*Neutral*	0.5445	*0.6051*	0.5957	0.50
	Optimist	0.1879	0.2851	*0.3371*	0.1532
	Pessimist	*0.8741*	0.8625	0.7962	0.8468

Table 5.9. The ranking results

Methods	Targets	Ranking
Normalization	*Neutral*	$A_2 \succ A_3 \succ A_1 \succ A_2$
	Optimist	$A_3 \succ A_2 \succ A_1 \succ A_4$
	Pessimist	$A_2 \succ A_1 \succ A_4 \succ A_3$
Alpha-Cut	*Neutral*	$A_2 \succ A_3 \succ A_1 \succ A_4$
	Optimist	$A_3 \succ A_2 \succ A_1 \succ A_4$
	Pessimist	$A_1 \succ A_2 \succ A_4 \succ A_3$

the ranking order of alternatives corresponding to the neutral target is the same as that obtained by using the fuzzy expected profits with centroid-based ranking criterion, where the risk neutrality is assumed. For the case of optimistic target T_{opt}, as discussed above, the DM has a risk-seeking behavior and then he wishes to have profit as big as possible accepting a risk that if the desirable state will not occur, he may get a big loss. For both models, this attitude leads to the selection of alternative A_3 which has the biggest profit in case of a high demand occurs. By the contrast, the pessimistic target T_{pess} corresponds to a risk-aversion behavior of the DM. As we have seen, in this case the alternative A_3 becomes the worst and alternatives A_1 and A_2 are more preferred than the others in both models. This reflects the situation that the DM is looking for sure of getting profit. However, while the alpha-based model suggests the selection of A_1, the normalization-based model still ranks A_2 as the best. This again exhibits the same behavior, as observed above, in that the value function in the alpha-based model reflects a stronger decision attitude towards the target than that in the normalization-based model.

5.6 Conclusion

In this chapter, we have introduced how to bring fuzzy targets within the reach of DUU paradigm on which an interesting link between different attitudes about target and different risk attitudes in terms of utility functions has been established. Furthermore, it has been also shown that the fuzzy target-based approach would provide an appealing and unified one for fuzzy decision analysis with uncertainty, which allows for including the DM's attitude expressed in the target assessment into the formulation of target-based decision functions.

By the introduction of a fuzzy target-based approach to DUU in this chapter, we think that it suggests an interesting perspective for further studies on various different decision problems. The first problem of constructing target-based decision functions for attitudinal decision making with uncertainty as an extension of that developed recently by Yager (1999) would be worth to be studied. Also, it would be interesting to study whether and how a fuzzy target-based approach can be applied to developing decision models for multicriteria decision making (Dubois et al. 2000) as well as group decision making. For example, by the structural relation between decision under uncertainty and multicriteria decision making models established in (Fortemps and Roubens 1996), we could straightforwardly apply the fuzzy target-based model to multicriteria decision making in a similar way as that introduced for DUU, however, further research is required.

References

A. E. Abbas, J. E. Matheson: Normative target-based decision making. Managerial and Decision Economics 26 (2005) 373–385

R. Bordley, M. LiCalzi: Decision analysis using targets instead of utility functions. Decisions in Economics and Finance 23(1) (2000) 53–74

R. Bordley: Foundations of target-based decision theory. In: U. Derigs (Ed.), Optimization and Operations Research, from Encyclopedia of Life Support Systems (EOLSS), Developed under the Auspices of the UNESCO, Eolss Publishers, Oxford, UK (2004) [http://www.eolss.net]

R. Bordley, C. Kirkwood: Multiattribute preference analysis with performance targets. Operations Research 52(6) (2004) 823–835

G. Bortolan, R. Degani: A review of some methods for ranking fuzzy subsets. Fuzzy Sets and Systems 15 (1985) 1–19

H. W. Brachinger, P.-A. Monney: Decision analysis. In: U. Derigs (Ed.), Optimization and Operations Research, from Encyclopedia of Life Support Systems (EOLSS), Developed under the Auspices of the UNESCO, Eolss Publishers, Oxford, UK (2004) [http://www.eolss.net]

E. Castagnoli, M. LiCalzi: Expected utility without utility. Theory and Decision 41(3) (1996) 281–301

E. Castagnoli, M. LiCalzi: Benchmarking real-valued acts. Games and Economic Behavior 57 (2006) 236–253

S. J. Chen, C. L. Hwang: Fuzzy Multiple Attribute Decision Making. Springer, Berlin Heidelberg New York (1992)

L. H. Chen, H. W. Lu: An approximate approach for ranking fuzzy numbers based on left and right dominance. Computer and Mathematics with Applications 41 (2001) 1589–1602

D. Dubois, H. Prade: Properties of measures of information in evidence and possibility theories. Fuzzy Sets and Systems 24 (1987) 161–182

D. Dubois, H. Prade: Fuzzy sets, possibility and measurement. European Journal of Operational Research 40(2) (1989) 135–154

D. Dubois, M. Grabisch, F. Modave, H. Prade: Relating decision under uncertainty and multicriteria decision making models. International Journal of Intelligent Systems 15(10) (2000) 967–979

P. Fortemps, M. Roubens: Ranking and defuzzication methods based on area compensation. Fuzzy Sets and Systems 82 (1996) 319–330

V. N. Huynh, Y. Nakamori, M. Ryoke, T. B. Ho: A fuzzy target based model for decision making under uncertainty. Proceedings of IEEE International Conference on Fuzzy Systems, Vancouver, Canada, July 2006a, pp. 1611–1618

V. N. Huynh, Y. Nakamori, J. Lawry: A probability-based approach to comparison of fuzzy numbers and applications to target-oriented decision making. IEEE Transactions on Fuzzy Systems [DOI: 10.1109/TFU22.2007.896315]

V. N. Huynh, Y. Nakamori, T. B. Ho: Utility function induced by fuzzy target in probabilistic decision making. In: S. Greco et al. (Eds.): RSCTC 2006, LNAI 4259, pp. 296–305, Springer-Verlag, Berlin (2006b)

D. Kahneman, A. Tversky: Prospect theory: An analysis of decision under risk. Econometrica 47(2) (1979) 263–291

G. J. Klir: Uncertainty and Information: Foundations of Generalized Information Theory. Wiley, New Jersey (2006)

H. Lee-Kwang, J.-H. Lee: A method for ranking fuzzy numbers and its application to decision-making. IEEE Transactions on Fuzzy Systems 7(6) (1999) 677–685

M. LiCalzi: A language for the construction of preferences under uncertainty. Revista de la Real Academia de Ciencias Exactas, Fisicas y Naturales 93 (1999) 439–450

C. F. Manski: Ordinal utility models of decision making under uncertainty. Theory and Decision 25 (1988) 79–104

H. T. Nguyen: A note on the extension principle for fuzzy sets. Journal of Mathematical Analysis and Applications 64 (1978) 369–380

H. J. Rommelfanger: Fuzzy decision theory. In: U. Derigs (Ed.), Optimization and Operations Research, from Encyclopedia of Life Support Systems (EOLSS), Developed under the Auspices of the UNESCO, Eolss Publishers, Oxford, UK (2004) [http://www.eolss.net]

J. Saade: A unifying approach to defuzzication and comparison of the outputs of fuzzy controllers. IEEE Transactions on Fuzzy Systems 4(3) (1996) 227–237

L. J. Savage: The Foundations of Statistics. Wiley, New York (1954)

G. Shafer: A Mathematical Theory of Evidence. Princeton University Press, Princeton (1976)

J. Von Neumann, O. Morgenstern: Theory of Games and Economic Behavior. Princeton University Press, Princeton (1944)

R. R. Yager. Fuzzy modeling for intelligent decision making under uncertainty, IEEE Transactions on Systems, Man and Cybernetics Part B 30(1) (2000) 60–70

R. R. Yager, M. Detyniecki, B. Bouchon-Meunier: A context-dependent method for ordering fuzzy numbers using probabilities. Information Sciences 138(1–4) (2001) 237–255

R. R. Yager: On the instantiation of possibility distributions. Fuzzy Sets and Systems 128(2) (2002) 261–266

R. R. Yager: Including decision attitude in probabilistic decision making. International Journal of Approximate Reasoning 21 (1999) 1–21

L. A. Zadeh: Fuzzy sets as a basic for a theory of possibility. Fuzzy Sets and Systems 1 (1978) 3–28

H.-J. Zimmerman: Fuzzy Set Theory and Its Applications. Kluwer-Nijhof, New York (1985)

6

An Approximation Kuhn–Tucker Approach for Fuzzy Linear Bilevel Decision Making

Guangquan Zhang, Jie Lu, and Tharam Dillon

Faculty of Information Technology, University of Technology, Sydney
PO Box 123, Broadway Sydney NSW 2007, Australia
zhangg@it.uts.edu.au, jielu@it.uts.edu.au, tharam@it.uts.edu.au

Summary. In bilevel decision making, the leader aims to achieve an optimal solution by considering the follower's optimized strategy to react each of his/her possible decisions. In a real-world bilevel decision environment, uncertainty must be considered when modeling the objective functions and constraints of the leader and the follower. Following our previous work, this chapter proposes a fuzzy bilevel decision making model to describe bilevel decision making under uncertainty. After giving the definitions of optimal solutions and related theorems for fuzzy bilevel decision problems this chapter develops an approximation Kuhn–Tucker approach to solve the problem. Finally, an example of reverse logistics management illustrates the application of this proposed fuzzy bilevel decision making approach.

6.1 Introduction

Organizational decision making often involves two decision levels. Decision maker at the upper level is termed as the leader, and at the lower level, the follower. When the leader attempts to optimize his/her objective, the follower tries to find an optimized strategy according to each of possible decisions made by the leader (Bard 1998; Bracken and McGill 1973). This is called a bilevel decision making problem.

Bilevel decision making (also called bilevel programming, BLP) techniques, introduced by Von Stackelberg (1952), have been developed for mainly solving decentralized decision problems with decision makers in a hierarchical organization (Anandalingam and Friesz 1992; Lai 1996). Bilevel decision making techniques have been applied with remarkable; White and Anandalingam 1993 success in different domains, for example, decentralized resource, planning, electronic power market, logistics, civil engineering, chemical engineering and road network management (Aiyoshi and Shimizu 1981; Amat and McCarl 1981; Leblanc and Boyce 1986; Marcotte 1983; Miller et al. 1992; Papavassilopoulos 1982). The majority of research on bilevel decision making has centered on the linear version of the problem (Candler and Townsley 1982;

G. Zhang et al.: *An Approximation Kuhn–Tucker Approach for Fuzzy Linear Bilevel Decision Making*, Studies in Computational Intelligence (SCI) **97**, 157–171 (2008)
www.springerlink.com

Chen et al. 1992; Dempe 1987). A set of approaches and algorithms of linear bilevel programming, such as well known Kuhn–Tucker approach (Bard 1998; Bard and Falk 1982), Kth-best approach (Bialas and Karwan 1984; Candler and Townsley 1982) and Branch-and-bound algorithm (Hansen et al. 1992), have obtained well applications. However, existing bilevel decision making approaches and algorithms mainly suppose the situation in which the objective functions and constraints are characterized with precise parameters. Therefore, the parameters in modeling a bilevel decision problem are required to be fixed at the some values in an experimental and/or subjective manner through the experts' understanding of the nature of the parameters. It has been observed that, in most real-world situations, for example, logistics, the possible values of these parameters are often only imprecisely or ambiguously known to the experts who establish this model. With this observation, it would be certainly more appropriate to interpret the experts' understanding of the parameters as fuzzy numerical data which can be represented by means of fuzzy sets (Zadeh 1965) A bilevel decision problem in which the parameters, either in objective functions or in constrains or both of the leader or the follower or both, are described by fuzzy values is called a fuzzy bilevel programming (FBLP) or a fuzzy bilevel decision making (FBLDM) problem.

Based on the definitions given by Bard (1982, 1998) and Lai (1996) a BLP problem can be described into two situations: cooperative and uncooperative. The cooperative BLP assumes the leader and the follower making their decisions under a cooperative relationship, while an uncooperative BLP problem assumes that the follower re-acts the leader's decision in a totally personally optimal way. Both situations can be happened in real world decision-making practice. The FBLP problem was first researched by Sakawa et al. (2000). Sakawa et al. has formulated a cooperative FBLP problem and proposed a fuzzy bilevel programming approach for solving the problem. In the approach, Sakawa introduced the concepts of α-bilevel programming based on the basis of fuzzy number α-level sets. Also, Shih et al. (1996) and Shih and Lee (1999) completed a related work but with different focuses. Shih proposed to use fuzzy approach to solve a bilevel decision problem. Under some fuzzy assumptions, the leader first defines his/her objective with some tolerances which are described by fuzzy membership functions. The follower then makes his/her decision based on the tolerances. Basically, the research mainly focuses on solving a non-fuzzy bilevel problem by using fuzzy sets techniques. Different from these results, this chapter focuses on when a bilevel decision model contains various uncertain parameters and is within an uncooperative situation. Based on the extended solution concept and related theorems of BLP proposed in (Shi et al. 2005a–c), we have first solved an FBLP problem with a triangular form of membership function in its fuzzy parameters (Zhang et al. 2003b; Zhang and Lu 2005). This chapter extends our previous research by allowing any form of membership functions to describe the fuzzy parameters in an FBLP model. It particularly develops an approximation Kuhn–Tucker approach to solve the general FBLP problem.

Following the introduction, Sect. 6.2 presents a model for fuzzy linear bilevel decision problem and related definitions, theorems and properties. A general fuzzy numbers based approximation Kuhn–Tucker approach for solving FBLP problems is presented in Sect. 6.3. A logistics management example is shown in Sect. 6.4 for illustrating the proposed approach. Conclusions and further study are discussed in Sect. 6.5.

6.2 A Model for Fuzzy Linear Bilevel Decision Problems

In this section we will propose a model for fuzzy linear bilevel decision problems. We will also give the way to formulate such problems and the necessary and sufficient condition for solving the problems.

We will first introduce a model and related solving approach for linear bilevel decision problems and then extend it to fuzzy linear bilevel decision problems.

For $x \in X \subset R^n$, $y \in Y \subset R^m$, $F : X \times Y \to R^1$, and $f : X \times Y \to R^1$, a linear BLP problem is given by Bard (1998):

$$\min_{x \in X} F(x, y) = c_1 x + d_1 y \tag{6.1a}$$

$$\text{subject to } A_1 x + B_1 y \leq b_1 \tag{6.1b}$$

$$\min_{y \in Y} f(x, y) = c_2 x + d_2 y \tag{6.1c}$$

$$\text{subject to } A_2 x + B_2 y \leq b_2, \tag{6.1d}$$

where c_1, $c_2 \in R^n$, d_1, $d_2 \in R^m$, $b_1 \in R^p$, $b_2 \in R^q$, $A_1 \in R^{p \times n}$, $B_1 \in R^{p \times m}$, $A_2 \in R^{q \times n}$, $B_2 \in R^{q \times m}$. $F(x, y)$ is the leader's objective function, and $f(x, y)$ is the follower's objective function. $A_1 x + B_1 y \leq b_1$ and $A_2 x + B_2 y \leq b_2$ are the constraints of the leader and the follower.

Let $u \in R^p$, $v \in R^q$ and $w \in R^m$ be the dual variables associated with constraints (6.1b), (6.1d) and $y \geq 0$, respectively. We have the following theorem.

Theorem 1. *A necessary and sufficient condition that (x^*, y^*) solves the linear BLP problem (1) is that there exist (row) vectors u^*, v^* and w^* such that $(x^*, y^*, u^*, v^*, w^*)$ solves:*

$$\min F(x, y) = c_1 x + d_1 y \tag{6.2a}$$

$$\text{subject to } A_1 x + B_1 y \leq b_1 \tag{6.2b}$$

$$A_2 x + B_2 y \leq b_2 \tag{6.2c}$$

$$u B_1 + v B_2 - w = -d_2 \tag{6.2d}$$

$$u(b_1 - A_1 x - B_1 y) + v(b_2 - A_2 x - B_2 y) + wy = 0 \tag{6.2e}$$

$$x \geq 0, y \geq 0, u \geq 0, v \geq 0, w \geq 0. \tag{6.2f}$$

Theorem 1 means that the most direct approach to solve (6.1) is to solve the equivalent mathematical program given in (6.2a–6.2f). One advantage that it offers is that it allows a more robust model to be solved without introducing any new computational difficulties.

Consider the following linear FBLP problem:

For $x \in X \subset R^n$, $y \in Y \subset R^m$, $F : X \times Y \to F^*(R)$, and $f : X \times Y \to F^*(R)$,

$$\min_{x \in X} F(x, y) = \tilde{c}_1 x + \tilde{d}_1 y \tag{6.3a}$$

$$\text{subject to } \tilde{a}_1 x + \tilde{b}_1 y \leqq \tilde{b}_1 \tag{6.3b}$$

$$\min_{y \in Y} f(x, y) = \tilde{c}_2 x + \tilde{d}_2 y \tag{6.3c}$$

$$\text{subject to } \tilde{a}_2 x + \tilde{b}_2 y \leqq \tilde{b}_2, \tag{6.3d}$$

where $\tilde{c}_1,\ \tilde{c}_2 \in F^*(R^n)$, $\tilde{d}_1,\ \tilde{d}_2 \in F^*(R^m)$, $\tilde{b}_1 \in F^*(R^p)$, $\tilde{b}_2 \in F^*(R^q)$, $\tilde{a}_1 = (\tilde{a}_{ij})_{p\times n}$, $\tilde{a}_{ij} \in F^*(R)$, $\tilde{b}_1 = \left(\tilde{b}_{ij}\right)_{p\times m}$, $\tilde{b}_{ij} \in F^*(R)$, $\tilde{a}_2 = (\tilde{e}_{ij})_{q\times n}$, $\tilde{e}_{ij} \in F^*(R)$, $\tilde{b}_2 = (\tilde{s}_{ij})_{q\times m}$, $\tilde{s}_{ij} \in F^*(R)$. Here, $F(x, y)$ and $f(x, y)$ are the fuzzy objectives of the leader and the follower.

Associated with the FBLP problem, we now consider the following multi-objective linear bilevel programming (MLBLP) problem:

For $x \in X \subset R^n$, $y \in Y \subset R^m$, $F : X \times Y \to F^*(R)$, and $f : X \times Y \to F^*(R)$,

$$\begin{aligned} &\min_{x \in X} (F(x,y))_\lambda^L = {c_1}_\lambda^L x + {d_1}_\lambda^L y, \lambda \in [0,1] \\ &\min_{x \in X} (F(x,y))_\lambda^R = {c_1}_\lambda^R x + {d_1}_\lambda^R y, \lambda \in [0,1] \end{aligned} \tag{6.4a}$$

$$\text{subject to } {A_1}_\lambda^L x + {B_1}_\lambda^L y \leqq {b_1}_\lambda^L, {A_1}_\lambda^R x + {B_1}_\lambda^R y \leqq {b_1}_\lambda^R, \lambda \in [0,1] \tag{6.4b}$$

$$\begin{aligned} &\min_{y \in Y} (f(x,y))_\lambda^L = {c_2}_\lambda^L x + {d_2}_\lambda^L y, \lambda \in [0,1] \\ &\min_{y \in Y} (f(x,y))_\lambda^R = {c_2}_\lambda^R x + {d_2}_\lambda^R y, \lambda \in [0,1] \end{aligned} \tag{6.4c}$$

$$\text{subject to } {A_2}_\lambda^L x + {B_2}_\lambda^L y \leqq {b_2}_\lambda^L, {A_2}_\lambda^R x + {B_2}_\lambda^R y \leqq {b_2}_\lambda^R, \lambda \in [0,1], \tag{6.4d}$$

where ${c_1}_\lambda^L$, ${c_1}_\lambda^R$, ${c_2}_\lambda^L$, ${c_2}_\lambda^R \in R^n$, ${d_1}_\lambda^L$, ${d_1}_\lambda^R$, ${d_2}_\lambda^L$, ${d_2}_\lambda^R \in R^m$, ${b_1}_\lambda^L$, ${b_1}_\lambda^R \in R^p$, ${b_2}_\lambda^L$, ${b_2}_\lambda^R \in R^q$, ${A_1}_\lambda^L = \left({a_{ij}}_\lambda^L\right)$, ${A_1}_\lambda^R = \left({a_{ij}}_\lambda^R\right) \in R^{p\times n}$, ${B_1}_\lambda^L = \left({b_{ij}}_\lambda^L\right)$, ${B_1}_\lambda^R = \left({b_{ij}}_\lambda^R\right) \in R^{p\times m}$, ${A_2}_\lambda^L = \left({e_{ij}}_\lambda^L\right)$, ${A_2}_\lambda^R = \left({e_{ij}}_\lambda^R\right) \in R^{q\times n}$, ${B_2}_\lambda^L = \left({s_{ij}}_\lambda^L\right)$, ${B_2}_\lambda^R = \left({s_{ij}}_\lambda^R\right) \in R^{q\times m}$.

Obviously, the leader and the follower each has two objective functions but the parameters of those functions are non-fuzzy values.

From the order relationship (Zhang et al. 2003a, b) of fuzzy numbers, we have the following Theorem 2, which will provide a solution to the fuzzy linear bilevel decision problems.

Theorem 2. *Let (x^*, y^*) be the solution of the MLBLP problem (6.4a–6.4d). Then it is also a solution of the FBLP problem defined by (6.3a–6.3d).*

Theorem 3 below will tell us that if all fuzzy parameters in a fuzzy linear bilevel decision problem have trapezoidal membership functions, then its optimization solution is equal to the optimization solution of the multi-objective linear bilevel decision problems.

Theorem 3. *For $x \in X \subset R^n$, $y \in Y \subset R^m$, if all the fuzzy parameters $\tilde{a}_{ij}$, $\tilde{b}_{ij}$, $\tilde{e}_{ij}$, $\tilde{s}_{ij}$, $\tilde{c}_i$ and $\tilde{d}_i$ have trapezoidal membership functions in the FBLP problem (6.3),*

$$\mu_{\tilde{z}}(t) = \begin{cases} 0 & t < z_\beta^L \\ \frac{\alpha-\beta}{z_\alpha^L - z_\beta^L}\left(t - z_\beta^L\right) + \beta & z_\beta^L \leqq t < z_\alpha^L \\ \alpha & z_\alpha^L \leqq t < z_\alpha^R \\ \frac{\alpha-\beta}{z_\beta^R - z_\alpha^R}\left(-t + z_\beta^R\right) + \beta & z_\alpha^R \leqq t \leqq z_\beta^R \\ 0 & z_\beta^R < t \end{cases}, \tag{6.5}$$

where $\tilde{z}$ denotes $\tilde{a}_{ij}$, $\tilde{b}_{ij}$, $\tilde{e}_{ij}$, $\tilde{s}_{ij}$, $\tilde{c}_i$ and $\tilde{d}_i$ respectively, then it is the solution of the problem (6.3) that $(x^*, y^*) \in R^n \times R^m$ satisfying

$$\begin{aligned} &\min_{x \in X} (F(x,y))_\alpha^L = c_1{}_\alpha^L x + d_1{}_\alpha^L y, \\ &\min_{x \in X} (F(x,y))_\alpha^R = c_1{}_\alpha^R x + d_1{}_\alpha^R y, \\ &\min_{x \in X} (F(x,y))_\beta^L = c_1{}_\beta^L x + d_1{}_\beta^L y, \\ &\min_{x \in X} (F(x,y))_\beta^R = c_1{}_\beta^R x + d_1{}_\beta^R y, \end{aligned} \tag{6.6a}$$

$$\begin{aligned} \text{subject to } & A_1{}_\alpha^L x + B_1{}_\alpha^L y \leqq b_1{}_\alpha^L, \\ & A_1{}_\alpha^R x + B_1{}_\alpha^R y \leqq b_1{}_\alpha^R, \\ & A_1{}_\beta^L x + B_1{}_\beta^L y \leqq b_1{}_\beta^L, \\ & A_1{}_\beta^R x + B_1{}_\beta^R y \leqq b_1{}_\beta^R, \end{aligned} \tag{6.6b}$$

$$\begin{aligned} &\min_{y \in Y} (f(x,y))_\alpha^L = c_2{}_\alpha^L x + d_2{}_\alpha^L y, \\ &\min_{y \in Y} (f(x,y))_\alpha^L = c_2{}_\alpha^L x + d_2{}_\alpha^L y, \\ &\min_{y \in Y} (f(x,y))_\beta^L = c_2{}_\beta^L x + d_2{}_\beta^L y, \\ &\min_{y \in Y} (f(x,y))_\beta^R = c_2{}_\beta^R x + d_2{}_\beta^R y, \end{aligned} \tag{6.6c}$$

$$\begin{aligned}\text{subject to } & A_{2\alpha}^{L}x + B_{2\alpha}^{L}y \leqq b_{2\alpha}^{L},\\ & A_{2\alpha}^{L}x + B_{2\alpha}^{L}y \leqq b_{2\alpha}^{L},\\ & A_{2\beta}^{L}x + B_{2\beta}^{L}y \leqq b_{2\beta}^{L},\\ & A_{2\beta}^{R}x + B_{2\beta}^{R}y \leqq b_{2\beta}^{R}.\end{aligned} \tag{6.6d}$$

Obviously, now the leader and the follower each has four objective functions, but all parameters in these functions are non-fuzzy values.

Using the following lemma, we can prove the theorem.

Lemma 1 (Zhang and Lu 2007). *If there is (x^*, y^*) such that $c_\alpha^L x + d_\alpha^L y \geqq c_\alpha^L x^* + d_\alpha^L y^*, c_\beta^L x + d_\beta^L y \geqq c_\beta^L x^* + d_\beta^L y^*$, $c_\alpha^R x + d_\alpha^R y \geqq c_\alpha^R x^* + d_\alpha^R y^*$ and $c_\beta^R x + d_\beta^R y \geqq c_\beta^R x^* + d_\beta^R y^*$, for any $(x, y)(0 \leq \beta < \alpha \leq 1)$ and fuzzy sets $\tilde{c}$ and $\tilde{d}$ on R have trapezoidal membership function:*

$$\mu_{\tilde{e}}(x) = \begin{cases} 0 & x < e_\beta^L \\ \frac{\alpha-\beta}{e_\alpha^L - e_\beta^L}\left(x - c_\beta^L\right) + \beta & e_\beta^L \leqq x < e_\alpha^L \\ \alpha & e_\alpha^L \leqq x \leqq e_\alpha^R \\ \frac{\alpha-\beta}{e_\alpha^R - e_\beta^R}\left(x - e_\beta^R\right) + \beta & e_\alpha^R < x \leqq e_\beta^R \\ 0 & e_\beta^R < x \end{cases}$$

then

$$c_\lambda^L x + d_\lambda^L y \geqq c_\lambda^L x^* + d_\lambda^L y^*,$$
$$c_\lambda^R x + d_\lambda^R y \geqq c_\lambda^R x^* + d_\lambda^R y^*,$$

for any $\lambda \in [\beta, \alpha]$.

From Theorem 3, we can very easy get the following corollary.

Corollary 1. *For $x \in X \subset R^n$, $y \in Y \subset R^m$, if all the fuzzy parameters $\tilde{a}_{ij}, \tilde{b}_{ij}, \tilde{e}_{ij}, \tilde{s}_{ij}, \tilde{c}_i$ and $\tilde{d}_i$ have trapezoidal membership functions in the FBLP problem (6.3),*

$$\mu_{\tilde{z}}(t) = \begin{cases} 0 & t < z_{\alpha_0}^L \\ \frac{\alpha_1-\alpha_0}{z_{\alpha_1}^L - z_{\alpha_0}^L}\left(t - z_{\alpha_0}^L\right) + \alpha_0 & z_{\alpha_0}^L \leqq t < z_{\alpha_1}^L \\ \frac{\alpha_1-\alpha_0}{z_{\alpha_2}^L - z_{\alpha_1}^L}\left(t - z_{\alpha_1}^L\right) + \alpha_1 & z_{\alpha_1}^L \leqq t < z_{\alpha_2}^L \\ \cdots & \cdots \\ \alpha & z_{\alpha_n}^L \leqq t < z_{\alpha_n}^R \\ \frac{\alpha_n-\alpha_{n-1}}{z_{\alpha_{n-1}}^R - z_{\alpha_n}^R}\left(-t + z_{\alpha_{n-1}}^R\right) + \alpha_{n-1} & z_{\alpha_n}^R \leqq t < z_{\alpha_{n-1}}^R \\ \cdots & \cdots \\ \frac{\alpha_0-\alpha_1}{z_{\alpha_1}^R - z_{\alpha_0}^R}\left(-t + z_{\alpha_0}^R\right) + \alpha_0 & z_{\alpha_1}^R \leqq t \leqq z_{\alpha_0}^R \\ 0 & z_{\alpha_0}^R < t \end{cases}, \tag{6.7}$$

where $\tilde{z}$ denotes $\tilde{a}_{ij}, \tilde{b}_{ij}, \tilde{e}_{ij}, \tilde{s}_{ij}, \tilde{c}_i$ and $\tilde{d}_i$ respectively, then, it is the solution of the problem (3) that $(x^*, y^*) \in R^n \times R^m$ satisfying

$$\begin{aligned} &\min_{x \in X} (F(x,y))_{\alpha_i}^L = c1_{\alpha_i}^L x + d1_{\alpha_i}^L y, i = 0, 1, \ldots, n \\ &\min_{x \in X} (F(x,y))_{\alpha_i}^R = c1_{\alpha_i}^R x + d1_{\alpha_i}^R y, i = 0, 1, \ldots, n \end{aligned} \tag{6.8a}$$

$$\begin{aligned} \text{subject to } &A1_{\alpha_i}^L x + B1_{\alpha_i}^L y \leqq b1_{\alpha_i}^L, i = 0, 1, \ldots, n \\ &A1_{\alpha_i}^R x + B1_{\alpha_i}^R y \leqq b1_{\alpha_i}^R, i = 0, 1, \ldots, n \end{aligned} \tag{6.8b}$$

$$\begin{aligned} &\min_{y \in Y} (f(x,y))_{\alpha_i}^L = c2_{\alpha_i}^L x + d2_{\alpha_i}^L y, i = 0, 1, \ldots, n \\ &\min_{y \in Y} (f(x,y))_{\alpha_i}^R = c2_{\alpha_i}^R x + d2_{\alpha_i}^R y, i = 0, 1, \ldots, n \end{aligned} \tag{6.8c}$$

$$\begin{aligned} \text{subject to } &A2_{\alpha_i}^L x + B2_{\alpha_i}^L y \leqq b2_{\alpha_i}^L, i = 0, 1, \ldots, n \\ &A2_{\alpha_i}^R x + B2_{\alpha_i}^R y \leqq b2_{\alpha_i}^R, i = 0, 1, \ldots, n \end{aligned} \tag{6.8d}$$

Now, we give the following important theorem by using Corollary 1, Theorem 1 and the method of weighting (Sakawa 1993). Based on this theorem, we will present an approximation approach for solving fuzzy linear bilevel decision problems.

Theorem 4. *For $x \in X \subset R^n$, $y \in Y \subset R^m$, if all the fuzzy parameters $\tilde{a}_{ij}$, $\tilde{b}_{ij}$, $\tilde{e}_{ij}$, $\tilde{s}_{ij}$, $\tilde{c}_i$ and $\tilde{d}_i$ have trapezoidal membership functions in the FBLP problem (6.3),*

$$\mu_{\tilde{z}}(t) = \begin{cases} 0 & t < z_{\alpha_0}^L \\ \frac{\alpha_1 - \alpha_0}{z_{\alpha_1}^L - z_{\alpha_0}^L}\left(t - z_{\alpha_0}^L\right) + \alpha_0 & z_{\alpha_0}^L \leqq t < z_{\alpha_1}^L \\ \frac{\alpha_1 - \alpha_0}{z_{\alpha_2}^L - z_{\alpha_1}^L}\left(t - z_{\alpha_1}^L\right) + \alpha_1 & z_{\alpha_1}^L \leqq t < z_{\alpha_2}^L \\ \cdots & \cdots \\ \alpha & z_{\alpha_n}^L \leqq t < z_{\alpha_n}^R \\ \frac{\alpha_n - \alpha_{n-1}}{z_{\alpha_{n-1}}^R - z_{\alpha_n}^R}\left(-t + z_{\alpha_{n-1}}^R\right) + \alpha_{n-1} & z_{\alpha_n}^R \leqq t < z_{\alpha_{n-1}}^R \\ \cdots & \cdots \\ \frac{\alpha_0 - \alpha_1}{z_{\alpha_1}^R - z_{\alpha_0}^R}\left(-t + z_{\alpha_0}^R\right) + \alpha_0 & z_{\alpha_1}^R \leqq t \leqq z_{\alpha_0}^R \\ 0 & z_{\alpha_0}^R < t \end{cases}, \tag{6.9}$$

where $\tilde{z}$ denotes $\tilde{a}_{ij}$, $\tilde{b}_{ij}$, $\tilde{e}_{ij}$, $\tilde{s}_{ij}$, $\tilde{c}_i$ and $\tilde{d}_i$ respectively, then a necessary and sufficient condition that (x^*, y^*) solves the FBLP problem (6.3) is that there exist (row) vectors u^*, v^* and w^* such that $(x^*, y^*, u^*, v^*, w^*)$ solves:

$$\min_{x \in X} (F(x,y)) = \sum_{i=0}^{n} \left(c1_{\alpha_i}^L x + d1_{\alpha_i}^L y\right) + \sum_{i=0}^{n} \left(c1_{\alpha_i}^R x + d1_{\alpha_i}^R y\right) \tag{6.10a}$$

$$\text{subject to } A_{1\alpha_i}^{L}x + B_{1\alpha_i}^{L}y \leqq b_{1\alpha_i}^{L}, i = 0, 1, \ldots, n \\ A_{1\alpha_i}^{R}x + B_{1\alpha_i}^{R}y \leqq b_{1\alpha_i}^{R}, i = 0, 1, \ldots, n \tag{6.10b}$$

$$A_{2\alpha_i}^{L}x + B_{2\alpha_i}^{L}y \leqq b_{2\alpha_i}^{L}, i = 0, 1, \ldots, n \\ A_{2\alpha_i}^{R}x + B_{2\alpha_i}^{R}y \leqq b_{2\alpha_i}^{R}, i = 0, 1, \ldots, n \tag{6.10c}$$

$$u\left(\sum_{i=0}^{n} B_{1\alpha_i}^{L} + \sum_{i=0}^{n} B_{1\alpha_i}^{R}\right) + v\left(\sum_{i=0}^{n} B_{2\alpha_i}^{L} + \sum_{i=0}^{n} B_{2\alpha_i}^{R}\right) - w \\ = -\left(\sum_{i=0}^{n} d_{2\alpha_i}^{L} + \sum_{i=0}^{n} d_{2\alpha_i}^{R}\right) \tag{6.10d}$$

$$u\left(\left(\sum_{i=0}^{n} b_{1\alpha_i}^{L} + \sum_{i=0}^{n} b_{1\alpha_i}^{R}\right) - \left(\sum_{i=0}^{n} A_{1\alpha_i}^{L} + \sum_{i=0}^{n} A_{1\alpha_i}^{R}\right)x \right. \\ \left. - \left(\sum_{i=0}^{n} B_{1\alpha_i}^{L} + \sum_{i=0}^{n} B_{1\alpha_i}^{R}\right)y\right) + v\left(\left(\sum_{i=0}^{n} b_{2\alpha_i}^{L} + \sum_{i=0}^{n} b_{2\alpha_i}^{R}\right)\right. \\ \left. - \left(\sum_{i=0}^{n} A_{2\alpha_i}^{L} + \sum_{i=0}^{n} A_{2\alpha_i}^{R}\right)x - \left(\sum_{i=0}^{n} B_{2\alpha_i}^{L} + \sum_{i=0}^{n} B_{2\alpha_i}^{R}\right)y\right) + wy = 0 \tag{6.10e}$$

$$x \geq 0, y \geq 0, u \geq 0, v \geq 0, w \geq 0. \tag{6.10f}$$

6.3 An Approximation Kuhn–Tucker Approach for Fuzzy Linear Bilevel Decision Problem

Based on the theories we proposed, we can find a way to solve a fuzzy bilevel decision making problem. We can first transform an original fuzzy bilevel decision making problem described in (6.3) into a multi-objective bilevel decision making problem where all parameters are not fuzzy values as described in (6.6). We then solve this nonfuzzy multi-objective bilevel decision making problem by repeating the use of Kuhn–Tucker approach until two solutions are very close to each other. This approach is described by the following six steps.

[Step 1] Transform the FBLP problem (6.3) to the MLBLP problem (6.6).
[Step 2] Let the interval $[0, 1]$ be decomposed into 2^{l-1} mean sub-intervals with $(2^{l-1} + 1)$ nodes λ_i $\left(i = 0, \ldots, 2^{l-1}\right)$ which are arranged in the order of $0 = \lambda_0 < \lambda_1 < \cdots < \lambda_{2^{l-1}} = 1$ and a range of errors $\varepsilon > 0$.
[Step 3] Set $l = 1$, then solve $(\text{MLBLP})_2^l$, i.e. (6.8) by using Kuhn–Tucker approach when $\beta = 0$ and $\alpha = 1$, we obtain an optimization solution $(x, y)_{2^l}$.
[Step 4] Solve $(\text{MLBLP})_2^{l+1}$ by Theorem 4 and Kuhn–Tucker approach. We obtain an optimization solution $(x, y)_{2^{l+1}}$.
[Step 5] If $\|(x, y)_{2^{l+1}} - (x, y)_{2^l}\| < \varepsilon$, then the solution (x^*, y^*) of the FBLP problem is $(x, y)_{2^{l+1}}$, otherwise, update l to $2l$ and go back to Step 3.

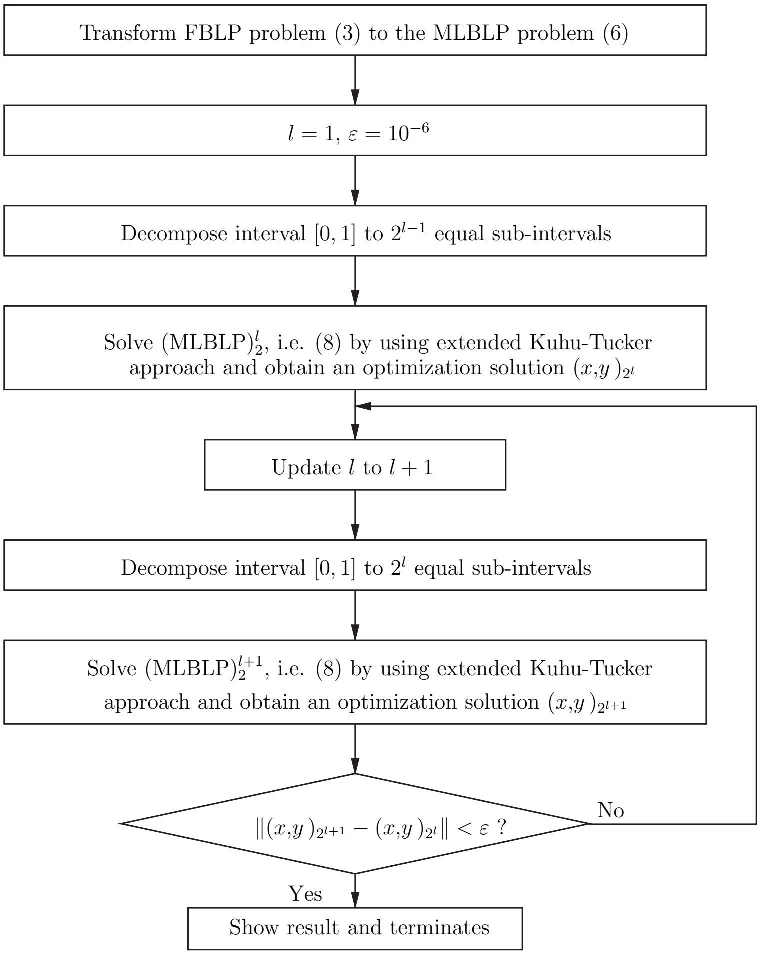


Fig. 6.1. A framework of the proposed approach

[Step 6] Show the solution.

Figure 6.1 shows a framework of the proposed approach. It has been implemented into a decision support system. Experiment shows it is very effective to solve a fuzzy bilevel decision making problem. In particular, the parameters can be any forms of fuzzy membership functions, such as triangular. An example will be given in Sect. 4 to illustrate the proposed approach.

6.4 An Application in Reverse Logistics Management

We apply the proposed approximation Kuhn–Tucker approach to support a bilevel decision making for a simple reverse logistics problem.

There are two logistics channels in a supply chain system. A forward logistics channel concerns the movement of goods from source to their point of consumption. A backward movement channel happens to return goods to suppliers called reverse logistics. In general, the forward logistics brings profits to all operational stages/departments involved, while a reverse logistics usually only brings costs. However, many companies have discovered that cost reductions in a reverse logistics can be substantial with an efficient planning and related strategies. An efficient planning for a reverse logistics chain involves two or multiple stages/departments of goods return process. Bilevel or multilevel decision-making approaches are therefore very promising to be applied in supporting such kind of decision-making.

Two main operational stages in a reverse logistics chain are the supplier and the distributor. They all aim to minimize their own cost but have individual constraints for a goods return. A decision about sharing cost of goods return made by the supplier will affect the decision made in the distributor, such as service quality provided to customers for a product return. Also, the distributor executes its policies after, and in view of, decisions made at the supplier stage. As the two stages in the chain are interrelated in a way that decisions made at one stage affect the performance of others, this can be seen as a bilevel decision making issue.

The supplier is the leader and the distributor is the follower in the decision issue. In almost cases of the real world, supplier and distributor each independently minimizes its cost on a reverse logistic chain but affected by each other. Furthermore, logistics managers in any stage of a logistics often imprecisely know the possible values of related costs. For example, they only can estimate possible inventory carrying cost and transportation cost of a particular set of goods to be returned. This situation brings about a demanding for the proposed bilevel decision making model of reverse logistics management to be able to handle uncertain information.

Let the supplier's objective function $\min_{x \in X} F(x, y)$ is to minimize the cost increasing by introducing online selling. The constraints of the supplier include the requirements of customer service and environment pollution issue. The distributor, as the followers, attempts to minimize their cost from the reverse logistics $\min_{y \in Y} f(x, y)$ for each policy made by the supplier.

When modeling the bilevel decision problem, the parameters for the objectives and constraints of both the leader and the follower are given by some uncertain experiment data and statistic reports from logistics managers. Therefore a fuzzy number based bilevel decision model is created for the reverse logistics decision problem. In order to easily show the application of the proposed approximation Kuhn–Tucker approach, the bilevel decision model is established by simplifying it into the following FBLP model.

Consider the following FBLP problem with $x \in R^1$, $y \in R^1$, and $X = \{x \geq 0\}$, $Y = \{y \geq 0\}$,

$$\text{Supplier}: \min_{x\in X} F(x,y) = \tilde{1}x + \tilde{2}y$$
$$\text{subject to } -\tilde{1}x + \tilde{3}y \leq \tilde{4}$$
$$\text{Distributor}: \quad \min_{y\in Y} f_1(x,y) = \tilde{1}x + \tilde{3}y$$
$$\text{subject to } \tilde{1}x - \tilde{1}y \leq \tilde{0}$$
$$-\tilde{1}x - \tilde{1}y \leq \tilde{0}\,,$$

where

$$\mu_{\tilde{1}}(t) = \begin{cases} 0 & t < 0 \\ t^2 & 0 \leqq t < 1 \\ 2-t & 1 \leqq t < 2 \\ 0 & 2 \leqq t \end{cases}, \quad \mu_{\tilde{2}}(t) = \begin{cases} 0 & t < 1 \\ t-1 & 1 \leqq t < 2 \\ 3-t & 2 \leqq t < 3 \\ 0 & 3 \leqq t \end{cases},$$

$$\mu_{\tilde{3}}(t) = \begin{cases} 0 & t < 2 \\ t-2 & 2 \leqq t < 3 \\ 4-t & 3 \leqq t < 4 \\ 0 & 4 \leqq t \end{cases}, \quad \mu_{\tilde{4}}(t) = \begin{cases} 0 & t < 3 \\ t-3 & 3 \leqq t < 4 \\ 5-t & 4 \leqq t < 5 \\ 0 & 5 \leqq t \end{cases},$$

$$\mu_{\tilde{0}}(t) = \begin{cases} 0 & t < -1 \\ t+1 & -1 \leqq t < 0 \\ 1-t^2 & 0 \leqq t < 1 \\ 0 & 1 \leqq t \end{cases}.$$

We now solve this problem by using the proposed approximation Kuhn–Tucker approach.

[Step 1] The FBLP problem is first transformed to the following MLBLP problem by using Theorem 2

$$\min_{x\in X} (F(x,y))_\lambda^L = \tilde{1}_\lambda^L x + \tilde{2}_\lambda^L y, \lambda \in [0,1]$$
$$\min_{x\in X} (F(x,y))_\lambda^R = \tilde{1}_\lambda^R x + \tilde{2}_\lambda^R y, \lambda \in [0,1]$$
$$\text{subject to } (-\tilde{1})_\lambda^L x + \tilde{3}_\lambda^L y \leqq \tilde{4}_\lambda^L, (-\tilde{1})_\lambda^R x + \tilde{3}_\lambda^R y \leqq \tilde{4}_\lambda^R, \lambda \in [0,1]$$
$$\min_{y\in Y} (f(x,y))_\lambda^L = \tilde{1}_\lambda^L x + \tilde{3}_\lambda^L y, \lambda \in [0,1]$$
$$\min_{y\in Y} (f(x,y))_\lambda^R = \tilde{1}_\lambda^R x + \tilde{3}_\lambda^R y, \lambda \in [0,1]$$
$$\text{subject to } \tilde{1}_\lambda^L x + (-\tilde{1})_\lambda^L y \leqq \tilde{0}_\lambda^L, \tilde{1}_\lambda^R x + (-\tilde{1})_\lambda^R y \leqq \tilde{0}_\lambda^R, \lambda \in [0,1]$$
$$(-\tilde{1})_\lambda^L x + (-\tilde{1})_\lambda^L y \leqq \tilde{0}_\lambda^L, (-\tilde{1})_\lambda^R x + (-\tilde{1})_\lambda^R y \leqq \tilde{0}_\lambda^R, \lambda \in [0,1]\,.$$

[Step 2] Let the interval $[0,1]$ be decomposed into 2^{l-1} mean sub-intervals with ($2^{l-1}+1$) nodes $\lambda_i \left(i=0,\ldots,2^{l-1}\right)$ which is arranged in the order of $0=\lambda_0<\lambda_1<\cdots<\lambda_{2^{l-1}}=1$ and a range of errors $\varepsilon=10^{-6}>0$.

[Step 3] When $l=1$, we solve the following MLBLP problem

$$
\begin{aligned}
&\min_{x\in X}(F(x,y))_1^{L(R)} = 1x+2y\\
&\min_{x\in X}(F(x,y))_0^{L} = 0x+y\\
&\min_{x\in X}(F(x,y))_0^{R} = 2x+3y\\
&\text{subject to } -1x+3y\le 4\\
&\qquad -2x+2y\le 3\\
&\qquad 0x+4y\le 5\\
&\qquad \min_{y\in Y}(f(x,y))_1^{L(R)} = 1x+3y\\
&\qquad \min_{y\in Y}(f(x,y))_1^{L} = 0x+2y\\
&\qquad \min_{y\in Y}(f(x,y))_0^{R} = 2x+4y\\
&\qquad \text{subject to } 1x-1y\le 0\\
&\qquad\qquad 0x-2y\le -1\\
&\qquad\qquad 2x-0y\le 1\\
&\qquad\qquad -1x-1y\le 0\\
&\qquad\qquad -2x-2y\le -1.
\end{aligned}
$$

The result is

$$
\begin{aligned}
&\min_{x\in X}(F(x,y))_1^{L(R)} = 1x+2y=3\\
&\min_{x\in X}(F(x,y))_0^{L} = 0x+y=1.25\\
&\min_{x\in X}(F(x,y))_0^{R} = 2x+3y=4.75\\
&\min_{y\in Y}(f(x,y))_1^{L(R)} = 1x+3y=4.25\\
&\min_{y\in Y}(f(x,y))_0^{L} = 0x+2y=2.5\\
&\min_{y\in Y}(f(x,y))_0^{R} = 2x+4y=6\\
&x=0.5, y=1.25.
\end{aligned}
$$

[Step 4] When $l = 2$, we solve it and get result.

$$
\begin{aligned}
&\min_{x\in X} F(x,y)_1^{L(R)} = 3\\
&\min_{x\in X} F(x,y)_{\frac{1}{2}}^{L} = 2.2286\\
&\min_{x\in X} F(x,y)_0^{L} = 1.25\\
&\min_{x\in X} F(x,y)_{\frac{1}{2}}^{R} = 3.875\\
&\min_{x\in X} F(x,y)_0^{R} = 4.75\\
&\min_{y\in Y} f(x,y)_1^{L(R)} = 4.25\\
&\min_{y\in Y} f(x,y)_{\frac{1}{2}}^{L} = 3.4786\\
&\min_{y\in Y} f(x,y)_1^{L} = 2.5\\
&\min_{y\in Y} f(x,y)_{\frac{1}{2}}^{R} = 5.125\\
&\min_{y\in Y} f(x,y)_0^{R} = 6\\
&x = 0.5, y = 1.25.
\end{aligned}
$$

[Step 5] $x = 0.5$, $y = 1.25$ is the optimal solution for the example because $\|(x,y)_{2^2} - (x,y)_{2^1}\| = 0 < \varepsilon$.

[Step 6] The optimization solution of the problem is $x = 0$, $y = 0.5$ such that

$$
\begin{aligned}
\text{Supplier :} \quad & \min_{x\in X} F(x,y) = \frac{\tilde{1}}{2} + \frac{5}{4}\cdot\tilde{2}\\
\text{Distributor :} \quad & \min_{y\in Y} f_1(x,y) = \frac{\tilde{1}}{2} + \frac{5}{4}\cdot\tilde{3}.
\end{aligned}
$$

This result tells us that when $x = 0.5$ and $y = 1.25$ the distribution can obtain the minimized cost around 4.25 and the supplier can get his/her minimized cost around 3.

This example shows how the approximation Kuhn–Tucker approach is used to solve an FBLP problem of a reverse logistic decision making.

6.5 Conclusion and Further Study

Bilevel decision problems often appear in organizational management activities, and involve various uncertainties. Therefore, fuzzy parameters based bilevel decision models can be more suitable to describe a real world bilevel decision situation. This chapter proposes a general fuzzy number based approximation Kuhn–Tucker approach to solve fuzzy bilevel decision problems. A logistics management example is given to illustrate the proposed approach.

Further study on this topic includes the development of a model and related approaches for fuzzy bilevel multi-follower decision problems. In such a kind of problems, multiple followers are involved a bilevel decision making activity. The leader's decision will be affected not only by those followers' individual reactions but also by the relationships among these followers. As uncertain data could occur in the objectives and constraints of both the leader and the multiple followers, it will be a challenge to get an optimal solution for the leader in the complex environment.

Acknowledgments

The work presented in this paper was supported by Australian Research Council (ARC) under discovery grants DP0557154.

References

Aiyoshi E. and Shimizu K. (1981) Hierarchical decentralized systems and its new solution by a barrier method, IEEE Transactions on Systems, Man, and Cybernetics 11: 444–449

Amat J.F. and McCarl B. (1981) A representation and economic interpretation of a two-level programming problem, Journal of the Operational Research Society 32: 783–792

Anandalingam G. and Friesz T. (1992) Hierarchical optimization: An introduction, Annals of Operations Research 34: 1–11

Bard J. (1998) Practical Bilevel Optimization: Algorithms and Applications, Amsterdam, Kluwer

Bard J. and Falk J. (1982) An explicit solution to the programming problem, Computers and Operations Research 9: 77–100

Bialas W. and Karwan M. (1984) Two-level linear programming, Management Science 30: 1004–1020

Bracken J. and McGill J. (1973) Mathematical programs with optimization problems in the constraints, Operations Research 21: 37–44

Candler W. and Townsley R. (1982) A linear two-level programming problem, Computers and Operations Research 9: 59–76

Chen Y., Florian M. and Wu S. (1992) A descent dual approach for linear bilevel programs, Technical Report CRT-866, Centre de Recherche sur les Transports

Dempe S. (1987) A simple algorithm for the linear bilevel programming problem, Optimization 18: 373–385

Hansen P., Jaumard B. and Savard G. (1992) New branch-and-bound rules for linear bilevel programming, SIAM Journal on Scientific and Statistical Computing 13: 1194–1217

Lai Y.J. (1996) Hierarchical optimization: A satisfactory solution, Fuzzy Sets and Systems 77: 321–335

Leblanc L. and Boyce D. (1986) A bilevel programming algorithm for exact solution of the network design problem with user-optimal flows, Transportation Research 20: 259–265

Marcotte P. (1983) Network optimization with continuous control parameters, Transportation Science 17: 181–197

Miller T., Friesz T. and Tobin R. (1992) Heuristic algorithms for delivered price spatially competitive network facility location problems, Annals of Operations Research 34: 177–202

Papavassilopoulos G. (1982) Algorithms for static Stackelberg games with linear costs and polyhedral constraints, Proceedings of the 21st IEEE Conference on Decisions and Control 647–652

Sakawa M. (1993) Fussy Sets and Interactive Mulitobjective Optimization, New York, Plenum

Sakawa M., Nishizaki I. and Uemura Y. (2000) Interactive fuzzy programming for multilevel linear programming problems with fuzzy parameters, Fuzzy Sets and Systems 109: 3–19

Shi C., Lu J. and Zhang G. (2005a) An extended Kuhn–Tucker approach for linear bilevel programming, Applied Mathematics and Computation 162: 51–63

Shi C., Lu J. and Zhang G. (2005b) An extended Kth-bast approach for linear bilevel programming, Applied Mathematics and Computation 164: 843–855

Shi C., Zhang G. and Lu J. (2005c) On the definition of linear bilevel programming solution, Applied Mathematics and Computation 160: 169–176

Shih H.S., Lai Y.J. and Lee E.S. (1996) Fuzzy approach for multilevel programming problems, Computers and Operations Research 23: 73–91

Shih H.S. and Lee E.S. (1999) Fuzzy multi-level minimum cost flow problems, Fuzzy Sets and Systems 107: 159–176

Von Stackelberg H. (1952) Theory of the Market Economy, New York, Oxford, Oxford University Press

White D. and Anandalingam G. (1993) A penalty function approach for solving bilevel linear programs, Journal of Global Optimization 3: 397–419

Zadeh L.A. (1965) Fuzzy sets, Information and Control 8: 338–353

Zhang G., Lu J. and Wu Y. (2003a) Formulation of linear programming problems with fuzzy coefficients of objective functions and constraints, Asian Information-Science-Life 2: 57–68

Zhang G., Wu Y., Remia M. and Lu J. (2003b) Formulation of fuzzy linear programming problems as four-objective constrained problems, Applied Mathematics and Computation 139: 383–399

Zhang G. and Lu J. (2005) The definition of optimal solution and an extended Kuhn–Tucker approach for fuzzy linear bilevel programming, The IEEE Intelligence Informatics Bulletin 5: 1–7

Zhang G. and Lu J. (2007) Model and approach of fuzzy bilevel decision making for logistics planning problem, Journal of Enterprise Information Management 20: 178–197

7

A Replanning Support for Critical Decision Making Situations: A Modelling Approach

Guy Camilleri, Jean-Luc Soubie, and Pascale Zarate

IC3 Group, IRIT, 118 route de Narbonne, 31062 Toulouse Cedex, France
{camiller,soubie,zarate}@irit.fr

Summary. This paper proposes principles for designing a tool aiding decision makers in critical situations defined by their unforeseeable occurrences and their potentially dramatic consequences. The purpose of this work consists of a modelling approach for Cooperative Knowledge Based Systems. This approach is based on a Task/Method paradigm that we describe. We then propose to take into account collateral effects of tasks in order to support decision makers in a critical context. The proposed tool should be able to generate a degraded solution for which collateral effects of previous tasks have been taken as new goals to reach. This study is possible thanks to the definition of several kinds of models: a Good Functioning Model and a Degraded Tasks Library.

7.1 Introduction

The purpose of this work focuses on decision making situations for which the context of the decision to make is completely unusual, i.e. with missing parameters or components out of service. The random factor takes, in this kind of situation, a very important dimension. Drucker in 1967 shows that there is no effective personality but that effective executives differ among each other in the same ways that ineffective ones do and this is of course the case for the decisions that have to be made out of the limits of nominal situations, i.e. foreseeable during the design process. We could define these situations as a badly defined context. Contingency management, in particular the management of unanticipated events, in the last 50 years, becomes an important and frequently debated issue in the scientific literature on complex systems management under risk conditions. Conventionally, planning the future presupposes collecting information and analysing it rationally in order to control for unexpected contingency events. These kinds of situations are called for us critical situations or non nominal situations. The idea of this work is no longer to support a decision maker in his classical decisional cognitive process but to assist him for an unusual decisional process. In order to act more

G. Camilleri et al.: *A Replanning Support for Critical Decision Making Situations: A Modelling Approach*, Studies in Computational Intelligence (SCI) **97**, 173–192 (2008)
www.springerlink.com

effectively, the objective of this work is to correctly anticipate the occurrence of extraordinary events.

The problem is no longer to make the most satisfying decision but to make the 'less worst decision'. The objective of the decision maker is no longer to reach the satisfactory solution but to minimize the bad effects of the event that led to the current situation. The idea is here to support the decision maker in a more proactive way. Some systems have been developed based on multi-criteria decision analysis and fuzzy set theory, as a useful learning tool for the governance of complex dynamic systems (for more details see Torrieri et al. 2002) but are not really effective for critical decision making situations. Our objective is to propose the basic principles for designing a system able to support these situations. In order to design such a system some constraints must be respected. The system must compute in a very rapid way the effects of a decision and propose solutions for which the effects of the critical situations are minored.

The knowledge models used to design usual decision support systems don't take into account critical situations following unforeseen events (see Little 1970 and also Camilleri et al. 2003b). In this case, the system becomes not only ineffective, but really dangerous for non expert users. Nevertheless, some usable pieces of modelled knowledge could give a very precious help in these critical situations. In order to benefit the most from modelled knowledge, we propose a planning approach that uses the models of low level tasks to build a degraded solution for the new problem. The purpose is to develop a system capable to generate new solutions by a replanning approach. In order to find a better solution, the effects of a previously proposed solution are taken into account.

In order to reach this objective, the Cooperative Knowledge Based Systems (CKBS) architecture is proposed. The CKBS is based on a global architecture in which several models are developed: user model, cooperation model and application model. This application model aims to represent the task to be achieved by the user. It is based on the task/method paradigm. This paradigm proposes to model classical situations in the form of tasks to do and associated methods to realize them.

According to the CKBS architecture, we actually study propose an adaptation of the initial formalism for such decision support systems specifically dedicated to critical situations.

The proposed approach is illustrated by an example of a very simplified situation. The context is about a plane piloting task. The pilot wants to land at the end of his flight but is informed about a failure with the landing gear. The pilot faces a critical situation in which he has to make a decision: land or not? Where?

The chapter is organised as follows. In the second section we briefly describe the Cooperative Knowledge Based System architecture and the role of its components. In the third section we develop our paradigm: the Task/Method paradigm for modelling situations; we then propose a model for the chosen example: the landing model. For the fifth section we propose to model the collateral effects and take those into account in order to replan a solution that will be proposed to the decision maker. Finally, we conclude and draw some perspectives of this work.

7.2 The Cooperative Knowledge Based Systems Architecture

Marakas in 2003 proposes an architecture of Knowledge Decision Support Systems composed by a data base management system, a model base management system, a man/machine interface and a knowledge engine. This architecture has the advantage to use the knowledge of decision makers. Nevertheless, the knowledge modelling still remains the problem of Knowledge Based systems. The proposed CKBS is based on the modelling step of Knowledge Based Systems design. The CKBS architecture is based on three main model libraries, each one corresponding to one representation required by the system to play its role in the cooperative problem solving:

- the application model (previously called domain model) represents the functional capabilities of the system in the domain of problem solving;
- the cooperation model represents the cooperative behaviour of the system with respect to the context and to the user;
- the user model is a user representation.

Another characteristic of our approach is to distinguish between the modelling level and the operational level of the system. The modelling level is previously described; at the operational level we describe the way how models are implemented:

- the application model is developed as a knowledge based implementation of the functional capabilities of the system;
- the cooperation model is operationalised in a knowledge base performing the high level control of the system, i.e. task assignment according to the organisational constraints, the state of the environment and the user's desires;
- the user model is imbedded in a user interface module.

The way to develop these models is based on the task-method paradigm. Models in CKBS theory are mainly handled by two kinds of processes: the execution process allowing the task achievement and the planning process that simulate the task performance.

7.3 The Task-Method Paradigm

7.3.1 Definition

The system must be able to decompose the problem in sub-problems but must also be able to share and control the assignment of tasks to agents (human or software) (for more details on the definition of a cooperative system see Cohen & Levesque 1990 and Sprague & Carlson 1982).

This paradigm is based on the decomposition of objectives in sub-tasks allowing their performance. With each sub-task at least one method is associated in order to perform it. The problem to solve is then modelled as a hierarchical tree. Terminal nodes represent the last sub-tasks to perform.

The way of experts' reasoning is represented in our work by a Task/Method paradigm. A Task is defined by the following components:

Name: Task name
Par: Parameter list handled by the task
Objective: Task's Goal
Methods: List of methods achieving the task.

The field ***Name*** specifies the name of the task. The list of parameters ***Par*** represents the set of world objects (described in the domain model) handled by the task. The ***Objective*** describes the task goal as a state of the world after task performance. This field seams redundant, however in natural language a goal can be expressed in two different ways (for more details see Bratman 1990), by a verb (corresponding to the name of the task) or through a state (described in the objective of the task). Sometimes, these two modes of expression are useful for correctly specifying the task, to make the definition less ambiguous. In the case where one of these modes of expression is not required, some functions transforming a proposition into task (and vice versa: Operators Do(act) and Achieve(p) see Grosz & Kraus 1996, Done(act) see Carberry 1988 or Occur(act) see Allen 1984) can be used. All methods defined during the modelling stage are recorded in the list of the task's ***Methods***.

For example: the task *transport_people_by_plane* can be described in the following way:

Name: *transport_people_by_plane*
Par: start: starting point, dest: destination point, ac: aircraft, p: passenger_set
Objective: at(p,dest)
Methods: *transport_people_by_plane*

A method describes a way (at only one level of abstraction) of carrying out a task.

A method is characterized by the following fields:

Name: Method name
Heading: Task achieved by the method

App-cond: Applicability conditions
Prec.: Preconditions that must be satisfied to be able to apply the method.
Effects: Effects generated by the successful application of the method.
Control: Achievement order of the sub-tasks
Sub-tasks: Sub-tasks set.

The task carried out by the method is indicated in the ***Heading***. The applicability conditions ***App-cond*** (as the task parameters) are used as constraints for the instantiation of the method. For example, the parameters start and dest of the task *transport_people_by_plane* (previously proposed) must be different; this constraint will be modelled under the applicability conditions by start $\neq$ dest. Thus, all instances of the method have different departure and arrival cities. The preconditions ***Prec*** are conditions that must be satisfied to apply the method. The difference between preconditions and applicability conditions is that an agent can satisfy the preconditions to apply the method, if an applicability condition is not satisfied then it will not try to satisfy it (for more details see Camilleri et al. 2003a). Applicability conditions can also be used to represent the conditions that cannot be satisfied (like: good weather or airport at cities, etc).

The ***Effects*** are caused by the application of the method carrying out the task. Some of the effects are present if and only if the task is carried out successfully. The task objectives necessarily belong to the effects, therefore all effects of all task's methods contain the task objective. Thus, all the methods performing a task must generate a state of the world containing the objective; the effects can nevertheless be different. The execution order of sub-tasks is described in the ***Control*** field, the sub-tasks are recorded in the field ***Sub-tasks***.

For example, the following method carries out the task transport:

Name: *transport_people_by_plane*
Heading: *transport_people_by_plane*
App-cond.: start $\neq$ dest, airport_at(start), airport_at(dest)
Prec.:
Effects: fuel_consumption, at(p, dest)
Sub-tasks: *takeoff, cruising, landing*

7.3.2 Task Achievement

Some tasks represented in the proposed Task-Method paradigm can be performed by an execution engine (see Cohen & Levesque 1990) that can be described in the following way:

```
Start(T: task)
if (the task T is a terminal task)
  then execute it
  else
```

```
  select applicable methods M of T
  found = false
  for all methods m in M and if not found do
  found = true
startMethod(m,found)/* which recalls this procedure
  according to the control field*/
  endfor
  if (not found) Failure
  endif
```

A terminal task is self performable. Its execution does not require any decomposition. Terminal tasks have only one method to achieve them, then the field Sub-tasks is empty and the control can point to an executable program.

The procedure start(t) runs the execution of the task t. In order to be carried out the task t must be instantiated giving a value to all parameters of the task. The execution engine performs the terminal task running the code attached to this task (procedure *startTerminal(t)*) else it uses a method to decompose the task. A method can be applied if and only if its preconditions and applicability conditions are satisfied. The execution engine chooses an applicable method and tries to carry out all subtasks of this method according to the control field (procedure *startMethod*). The procedure *start(t')* is then recursively applied on each sub-task. The execution process will stop if a subtask has no applicable method. The variable found will then be set to false (procedure *startMethod*).

7.3.3 The Process to Find a Solution

The Degraded Tasks Library

The use of CKBS as an intelligent decision support system (see Marakas 2003 and Zaraté 2005) implies a deep modelling of the process (see Newell 1982 and Soubie 1998) (problem solving, process control). The main model is a GFM (Good Functioning Model) that allows decision makers to face all kinds of situations. This model is a hierarchical decomposition of the main task which is to be supported. We propose in addition to the GFM to build a library containing methods for all the well known cases of critical situations, called the DTL (Degraded Task Library). This library can be used when foreseen events occur, but can also be used as a complement of the GFM by a planner to propose non-predefined methods.

The main model of the task can be called a GFM (Good Functioning Model). It is represented by a hierarchical decomposition of the task in sub-tasks. Then, if some degraded situations are well known, additional tasks can be modelled in the DTL (Degraded Tasks Library), in order to use them instead of the corresponding nominal ones.

Degraded tasks obviously have to be used only when critical situations occur. Such situations could be defined according to the good functioning model, by the lack of one or more applicability conditions for a task of this model. Thus, from the methodological point of view, the DTL consists mainly of tasks stemming from those of the GFM with one or more applicability conditions missing. The advantage to model the given problem in two libraries is that the DTL would be used only in critical situations, i.e. in very rare cases and the knowledge base would not take much more octets than the two libraries.

So, in the operational context, a plan is made to execute the main task through the selection and performance of a set of terminal tasks. When a foreseeable incident occurs, a new plan is calculated by taking into account the specific tasks.

Nevertheless, there are very critical situations where the methods of the initial model and the additional tasks cannot provide a solution in terms of the plan to apply. We propose in these cases, to use the structural characteristics of the tasks to find a new plan (new method). This plan is now based on the substitution of a "task objective" (initial goal) by a new goal corresponding to a set of effects of some methods of the task(s).

The Collateral Effects Modelling and Replanning Approach

Based on this principle the idea is now to model the other effects of the task. These effects are those for which the task is not necessarily designed. The collateral effects will be used in the replanning procedure. Another consequence of this choice is that the pre-conditions and applicability conditions of the tasks have to be clearly displayed to run the planning tool in a degraded context.

In our paradigm, a method is applicable if and only if its preconditions and applicability conditions are satisfied (hold in the world). Some degraded solutions cannot be treated by the methods described in the GFM (Good Functioning Model). Therefore in this last model, some preconditions of the methods are not present in these situations. The problem consists of completing a method of the GFM to satisfy the preconditions that do not hold, or to find a new method reaching the aimed goal.

In the following example one can see how to deal with critical situations using the task/method paradigm and the replanning approach.

7.4 A Case Study: The Landing Model

We present here a case study that is a simplified problem of aircraft landing. The presented model is obviously very incomplete. The tasks are presented in a very simplified version, only the tasks and subtasks names are given in Fig. 7.1

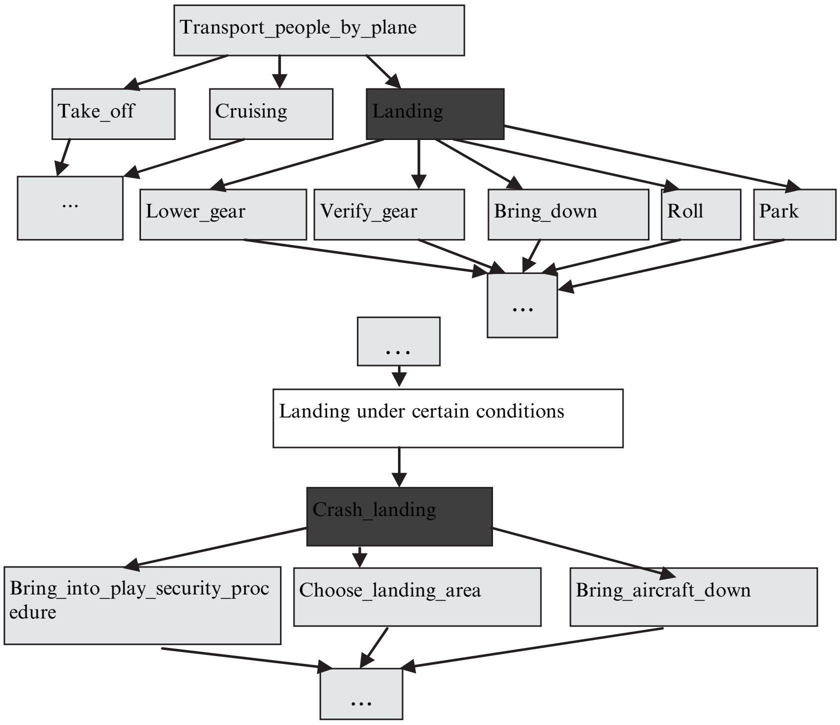


Fig. 7.1. The task/subtasks model

We present in Fig. 7.2 a part of the domain model thanks to a UML formalism. This model is a representation of the activity of piloting a plane.

The activity of piloting a plane is also modelled thanks to a tool that is developed by Camilleri et al. (see Clemen 1991), the Task Method Modelling Tool. This tool allows us to model the task to reach in an adequate formalism for the planning tool. We present in Fig. 7.3 a global model of the task to fulfil.

In the appendix A the same model is presented including the parameters, the collateral effects and the methods.

The specific task crash_landing is also modelled thanks to the TMMT tool in the Fig. 7.4.

In order to illustrate the use of replanning in case of a critical situation, consider now the above models for the *transport_people_by_plane* task.

In case of the impossibility of pulling down the landing gear, the landing task is not applicable (the applicability condition lg = OK is false). The planner first attempts to find a degraded task in the DTL model. The only available task is *crash_landing*. Indeed, this task shares with the landing task a part of the objective at(ac, ground). Nevertheless, the pre-condition of the method ft = empty is false at this moment.

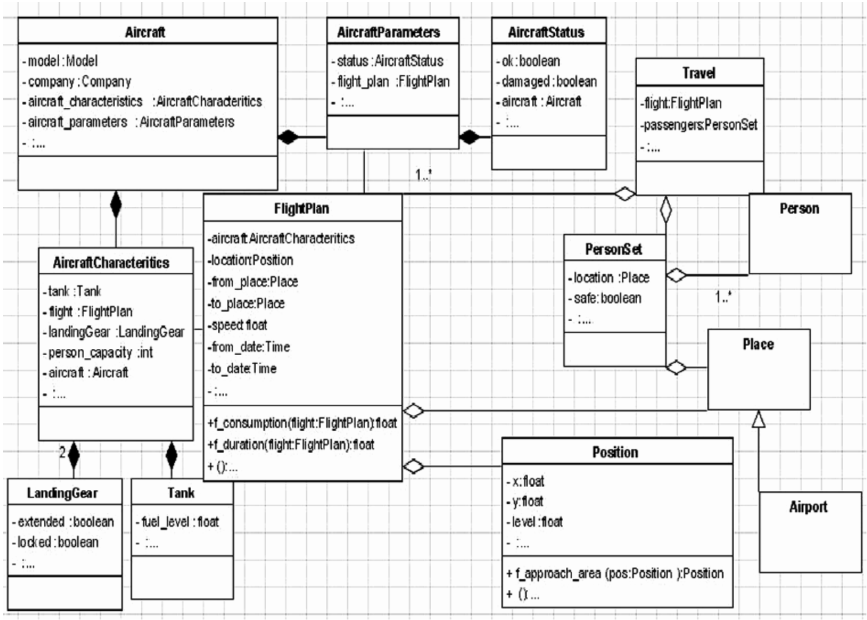


Fig. 7.2. The domain model thanks to the UML formalism

In order to satisfy the pre-condition of the only available method, the planner defines a new goal to reach: ft = empty. The implementation of this plan has to keep all the other pre-conditions of the *crash_landing* method. So, the planner goal includes all the pre-conditions of the method.

In the domain model, the planner finds the relation between ft = empty and fuel_consumption.

The planner considers all the effects of all the available methods and finds the two methods of the *cruising* task.

The first one implies that the aircraft keeps flying. The second one keeps the aircraft position (close to the airport).

So, the new method generated by the planner for the substitution of the *landing* task is: *cruising* using the *round* method, then *crash_landing* method.

The planner informs the user about the effects of this method:

damaged(ac), at(ac, ground), safe(p),

We show with this case study how the collateral effects are used and taken into account to find a degraded solution thanks to the Degraded Tasks Library. In this example the objective of the planner was no longer to support the pilot cruising but to find a solution for which the effect of the chosen method: *crash_landing* was minored. The system then proposes the pilot to make circle until the fuel tanker will be empty before landing. The idea is to keep the passengers in safety, as much as possible and to avoid an explosion, and this would be possible if the fuel tanker is empty. The system would be able to

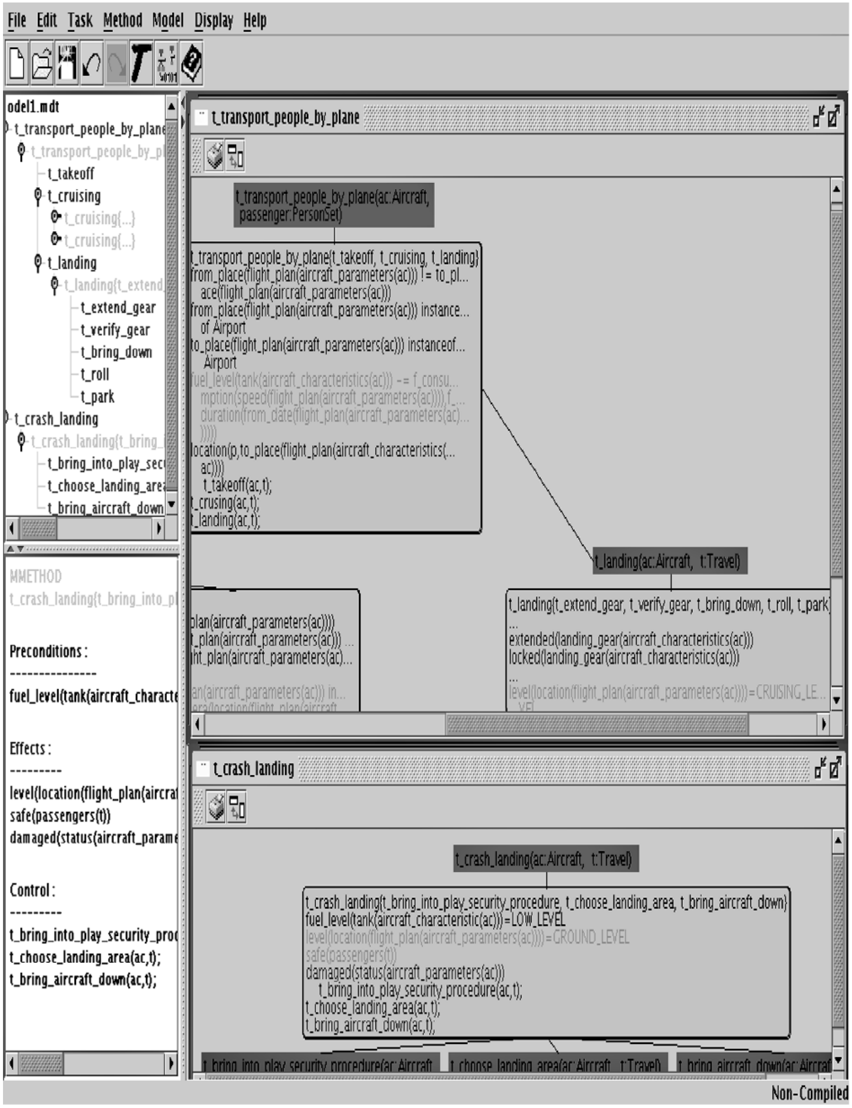


Fig. 7.3. The modelled task thanks to the task/method formalism

suggest that the pilot must join the departure point to the arrival point in x minutes and if the pilot does not acknowledge this choice the system will find another plan thanks to the replanning functionality in order to propose an alternate solution. This is the core of Intelligent DSS.

Fig. 7.4. The crash_landing method thanks the task/method formalism

7.5 Replanning Paradigm Through AI Planning Approaches

In the CKBS framework, before acting, the system must have a plan to satisfy goals. According to the current situation, plans are considered to be complete that is, their performances are assumed to reach goals from this current situation. During a plan execution, there are more ways (as environment and intention changes, actions execution failures, etc.) in which a plan can go wrong. In many occasions, the system needs to alter parts of its plan and possibly its goals to be able to adapt itself to a new environment (potentially a non-nominal case). In AI planning community, some works offer techniques called replanning to resolve execution failures of prior plan. These techniques appear suitable for plan management in intelligent decision making based on the CKBS architecture previously presented. In these works, the replanning process is regarded as an extension of plan generation (or planning) approaches (see van der Krogt & de Weerdt 2005).

In the first subsection, we briefly present the main approaches of plan generation in AI planning domain and their possible applications in model handling to dynamically build new methods (plans). Replanning techniques will be then exposed and finally we discuss the advantages of replanning processes for CKBS decision support systems.

7.5.1 Plan Generation Approaches in AI Planning

AI planning community aims at designing planning systems (called planners) that find a set of actions (plan) to reach a goal from a description of an initial state of the world. The major difference between planning processes and execution processes is: planning processes only determine plans allowing through its execution to reach a world state satisfying the aimed goal, while execution processes launch the achievement of actions that modify the world. As planning processes do not change the world, a description of the world changes is required. In this way, cancelling an applicable action is possible; moreover the cancelling action is often necessary in the planning process. However, in the execution process, coming back to previous states after an achievement action can be not possible.

Several complex problems (static vs. dynamic world, observable vs. partially observable world, etc.) are studied in the AI planning domain. Planning problems can be solved by two kinds of planning techniques; the action based planning and the hierarchical planning (see McDermott 2000). These two techniques can provide a useful help in models used for CKBS systems.

In the action based planning, the following elements are given: an initial state, a set of action definitions, and a proposition (goal) to be brought about. A solution (plan) is a set of actions that, when it is executed starting from the initial state, leads to a final state in which the goal is true. Recent advances in action-based planning concern: the improvement of the planner's efficiency GraphPlan (see Blum & Furst 1997), FF see Hoffmann & Nebel 2001, YAHSP (see Vidal 2004), FD (see Helmert 2006) etc.; the increase of tackled problems classes SAPA (see Do & Kambhampati Sapa 2001), FHP (see Zalaket & Camilleri 2004), TEG (see Baier & McIlraith 2006),

In addition to the material of action-based planning, hierarchical planning (also called knowledge-based planning by Wilkins and Desjardins 2001 uses a set of abstract actions. An abstract action cannot be applied directly, but must be applied by carrying out an expansion (or reduction) of it in terms of less abstract actions (terminal actions), typically that one finds in a "plan library". A problem may specify, in addition to a goal, an abstract action to be applied. A solution is a set of primitive actions that (a) achieves the goal; and (b) corresponds to an expansion of the given abstract action.

The modelling paradigm of plan library is action/method where one method represents a way to perform the action. The expansion process is often simple and effective. It is possible to describe in the plan library methods that contain different types of control (loops, conditions, etc.) and numerical knowledge. Moreover, a hierarchical representation is intuitive; users can easily interact with the planning process.

Both actions based planning and hierarchical planning can be seen as an iterative refinement of the set of all possible plans. This view is called refinement planning (see Kambhampati 1997) and shows that most existing planning algorithms can be conceived in this way.

In the next subsection, we will discuss the contributions and the adaptations of these two planning techniques to the model handling (see Zalaket & Camilleri 2004).

7.5.2 Advantages of Plan Generation Techniques in Model Handling

In several situations, particularly for decision making processes, it is often interesting to know that a task (or action) is applicable and its application will necessarily succeed. Hierarchical execution cannot guarantee the success of the execution before the task performance. As the execution process only checks the precondition of a method from the world state at one level of abstraction, it cannot know if there is for each task of this method another method that can be applied in the current situation. Hierarchical planning can be applied to cope with this problem. Hierarchical planners expand the action (or task) to find all primitive tasks reaching the goal; thus, they try several methods to find all methods of all actions. This process is possible because the action (or task) and the world state representation allow to avoid the performance of the action.

As previously mentioned, hierarchical models contain all applicable methods. For decision making processes, it is often difficult to model all methods of all the tasks for all situations, including critical ones. An action-based planner could be used to generate from a particular situation an applicable method. However, the current action-based planners are not able to solve the classes of problems suitable for the problem solving. For example, the current action-based planners cannot treat completely and efficiently numerical knowledge. Unfortunately, this kind of knowledge is frequently used in decision support systems design (see Keen & Scott Morton 1978, Klein & Methlie 1995 and Simon 1977). Thus, we try to design action-based planners that are adapted to the decision support. We implemented an action-based planner managing the numerical knowledge (see Zalaket & Camilleri 2004).

7.5.3 Replanning Techniques

Replanning techniques resolve execution failures of prior plans. These techniques are seen as an extension of planning algorithms. The majority of works (see van der Krogt & de Weerdt 2005) regard replanning as minimal perturbation or conservative planning (see Cushing & Kambhampati 2005, Nebel & Koehler 1995). Minimal perturbation planning starts from an initial (prior) plan and it tries to minimally alter the structure of this plan to build a new one which can be applied in the current situation. Van de Krogt and de Weerdt in 2005 propose a general minimal perturbation planning algorithm based on the refinement planning paradigm which unifies hierarchical and action based minimal perturbation planning. Many minimal perturbation planning works are interested in improving the planning efficiency, which in fact used minimal perturbation planning as a plan reuse paradigm. Minimal perturbation planning has in theory a greater complexity than classical plan generation (see Nebel & Koehler 1995), however in practice (with domain knowledge) it can be more efficient.

The Replan (see Blum & Furst 1997) module is a hierarchical minimal perturbation planner which uses a hierarchy of tasks (task networks) close to the proposed task/method paradigm. Initially, the replanner starts from the invalid leave (primitive task) of the current plan (tree); and then iteratively it goes up in the tree to find new applicable methods which form a new subtree. In the worst case, this process continues until the root of the hierarchy is reached, in this case the planning process is restarted from the scratch. The pruning heuristic used in the replanning component is encoded in a utility function. At each refinement, methods are chosen (or discarded) according to an interval of the expected utility. This replanner do not find the optimal plan that is, the plan which maximize the expected utility, but is inspired to the notion of persistence of intentions (see Bratman 1990) by trying to perform the most local changes. It is why this replanning component is a minimal perturbation planning.

However, the minimal perturbation planning has an important limitation (see Cushing & Kambhampati 2005), it does not take into account the stated intentions (commitments towards collaborators). Altering collaborator's intentions can degrade overall execution performance because a collaborator could have its own plan using these intentions. The quality of a replanning solution depends on the respect of commitments.

In the CKBS system, we propose to use a replanning approach without altering the commitments. A new plan is generated from the scratch by changing the expected utility in order to respect the commitments. In this way, the persistence of the intentions and the commitments are taken into account.

In some cases the replanning process allows one to face situations that have not been modelled (neither in the GFM nor the DTL). This becomes possible through a new methodology of modelling, taking into account not only the tasks goals, but also all the effects of their performance.

When the problem solver has no more solution using the classical planner, one can launch a replanning approach, starting from the current critical situation to reach a new degraded goal. This goal can be proposed by the user, and he also could specify the undesirable effects of the solution. In this case, the planner attempts to find a plan starting from the present world state. In order to reach this goal, it has to avoid the undesirable effects as much as possible (found in the effects of the methods in the GFM and DTL). When no other goal or effect is suggested by the decision maker, the planner brings the effect closer to the unreachable goal and adds all the method preconditions linked to this effect in order to build a new goal.

7.6 Conclusion

We have seen that cooperative knowledge based systems built on a set of task/method models could provide good intelligent decision support for nominal cases. Extending the hierarchical model to a library of subsets of degraded tasks model allows one to list rare well known incidents. Therefore, by means of a more complete modelling method, it is possible to provide a support that has not explicitly been foreseen at the system design stage, and a large flexibility in its use, that is essential in case of decision making in critical situations.

One perspective of this work is to develop a CKBS design methodology able to include these new features in terms of models and structure of the basic primitive task. Another perspective of this work is to expand the design of CKBS for several decision makers: expand it to group decision making (for more details about GDSS see Jessup & Valacich 1993).

Forgionne et al. in 2002 propose an I-DMSS architecture that can serve as a guideline to tailor system design and development for the specific decision problem. Starting with basic decision making support, such design and development may evolve through an ESS, IDSS or MSS into the general I-DMSS architecture as the decision-maker's needs and requirement mature. The idea of this work is to propose an intelligent decision support when decision makers are faced to critical situations, and finally when the problem is solved to come back to a nominal way of using the system.

References

Allen, J. (1984) Towards a general theory of action and time. Artificial Intelligence 23:123–154

Baier, J., and McIlraith, S. (2006) Planning with first-order temporally extended goals using heuristic search. In Proceedings of the Twenty first National Conference on Artificial Intelligence (AAAI06), pp 788–795, July, Boston, MA

Blum, A., and Furst, M. (1997) Fast planning through planning graph analysis. Artificial Intelligence 90(1–2):281–300

Boella, G., and Damiano, R. (2002) A replanning algorithm for a reactive agent architecture. In Artificial Intelligence: Methodology, Systems, and Applications (LLNCS 2443), 183–192. Springer

Bratman, M. (1990) What is Intention. In Cohen, P.R., Morgan, J.L., and Pollack, M.E. (eds.) Intentions in Communication.Cambridge MA: MIT, pp 15–31

Camilleri, G., Soubie, JL., and Zalaket, J. (2003a) TMMT: Tool Supporting Knowledge Modelling. In 7th International Conference on Knowledge-Based Intelligent Information and Engineering Systems, KES 2003, Springer, part I: 45–52

Camilleri, G., Soubie, JL., and Zaraté, P. (2003b) Critical Situations for Decision Making: A Support Based on a Modelling Tool. In Bui, T., Sroka, H., Stanek, S., and Gotuchowski, J. (eds.) DSS in the Uncertainty of the Internet Age. Katowice, pp 469–480

Carberry, S. (1988) Modelling the user's plans and goals. Computational Linguistics 14(3): 23–37

Clemen, R.T. (1991) Making hard decisions: An introduction to decision analysis. Belmont, CA: Duxbury

Cohen, P., and Levesque, H. (1990) Persistence, intention, and commitment. In Cohen, P.R., Morgan, J.L., and Pollack, M.E. (eds.) Intentions in Communication. Cambridge MA: MIT, pp 33–70

Cushing, W., and Kambhampati, S. (2005) Replanning: A new perspective. Poster Program. In Proceedings of the Fifteenth International Conference on Automated Planning and Scheduling (ICAPS-05)

Do, M., and Kambhampati Sapa, S. (2001) A domain-independent heuristic metric temporal planner. In Cesta, A., and Borrajo, D. (eds.) Proceedings of 6th European Conference on Planning. Springer, pp 109–120

Drucker, P. (1967) The Effective Executive. New York: Harper and Row

Forgionne, G., Mora, M., Cervantes, F., and 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., and Pomerol, J.C. (eds.) International Conference on Decision Making and Decision Support in the Internet Age (DSI-Age 2002), IFIP TC8/WG 8.3 Open Conference. University College of Cork (Cork, Ireland), pp 154–165

Grosz, B., and Kraus, S. (1996) Collaborative plans for complex group actions. Artificial Intelligence 86(2):269–357

Helmert, M. (2006) The fast downward planning system, Journal of Artificial Intelligence Research 26:191–246

Hoffmann, J., and Nebel, B. (2001) The FF planning system: Fast plan generation through heuristic search. Journal of Artificial Intelligence Research 14:253–302

Jessup, L., and Valacich, J. (eds.) (1993) Group support systems: New perspectives. New York: Macmillan

Kambhampati, S. (1997) Refinement planning as a unifying framework for plan synthesis. AI Magazine 18(2):67–97

Keen, P.G.W., and Scott Morton, M.S. (1978) Decision support systems: An organisational perspective. Reading, MA: Addison Wesley

Klein, M.R., and Methlie, L.B. (1995) Knowledge –based decision support systems. 2nd edition, New York: Wiley

Little, J.D. (1970) Models and managers: The concept of a Decision Calculus. Management Science 16(8) B4:66–85

Marakas, G.M. (2003) Decision support systems in the 21st Century. 2nd edition, Englewood Cliffs, NJ: Prentice Hall
McDermott, D. (2000) The 1998 AI planning systems competition. AI magazine 2(2):35–55
Nebel, B., and Koehler, J. (1995) Plan reuse versus plan generation: A theoretical and empirical analysis. Artificial Intelligence. 76(1–2):427–454
Newell, A. (1982) The knowledge level. Artificial Intelligence 18:87–127
Simon, H.A. (1977) The new Science of management decision. Englewood Cliffs, NJ: Prentice Hall
Soubie, J.L. (1998) Modelling in cooperative knowledge based systems. In Darses, F., Zaraté, P. (eds.) Proceedings of COOP'98 International Conference on Cooperative Systems, Cannes, France, part II, pp 45–48
Sprague, R.H., and Carlson, E.D. (1982) Building effective decision support systems. Englewood Cliffs, NJ: Prentice Hall
Torrieri, F., Concilio, G., and Nijkamp, P. (2002) Decision support tools for urban contingency policy: A scenario approach to risk management of the Vesuvio area in Naples, Italy. Journal of Contingencies and Crisis Management 10(2):95–112
van der Krogt, R., and de Weerdt, M. (2005) Plan repair as an extension of planning. In Proceedings of the Fifteenth International Conference on Automated Planning and Scheduling (ICAPS-05)
Vidal, V. (2004) A lookahead strategy for heuristic search planning, In Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling (ICAPS-2004), pp. 150–159, Whistler, Canada
Wilkins, D., and Desjardins, M. (2001) A call for knowledge-based planning. AI Magazine 22 (1):99–115
Zalaket, J., and Camilleri, G. (2004) FHP: Functional heuristic planning. In Negoita, M., Howlett, RJ., and Jain, L.C. (eds.) 8th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES 2004), Springer, part II, pp 9–16
Zaraté, P. (2005) Des Systèmes Interactifs d'Aide à la Décision aux Systèmes Coopératifs d'Aide à la Décision: Contributions conceptuelles et fonctionnelles, Mémoire d'habilitation à diriger des recherches, INPT, Décembre

Appendix A: Landing model

In this appendix a part of the landing model is presented thanks to TMMT tool. A syntax, coming from the C language, is used to describe comparison operators (equality==, not equal !=, . . .) in the preconditions and applicability conditions; and assignment operators ($=, + =, - =, \ldots$) in the objective and effect sets.

GFM (Good Functioning Model)

NON-TERMINAL TASK
t_transport_people_by_plane(ac:Aircraft, passenger:PersonSet)

Objectives:

location(passenger) = to_place(flight_plan(aircraft_characteristics(ac))))

Methods:

transport_people_by_plane{t_takeoff, t_cruising, t_landing}//flying nominal procedure

MMETHOD
transport_people_by_plane{t_takeoff, t_cruising, t_landing}

Applicability conditions:

from_place(flight_plan(aircraft_parameters(ac))) !=
to_place(flight_plan(aircraft_parameters(ac)))
from_place(flight_plan(aircraft_parameters(ac))) instanceof Airport
to_place(flight_plan(aircraft_parameters(ac))) instanceof Airport

Effects:

fuel_level(tank(aircraft_characteristics(ac))) − =
f_consumption(flight_plan(aircraft_parameters(ac))),
to_date(flight_plan(aircraft_parameters(ac))))+ =
f_duration(flight_plan(aircraft_parameters(ac))),
location(passenger) = to_place(flight_plan(aircraft_characteristics(ac))))

Control:

t_takeoff(ac,t);
t_crusing(ac,t);
t_landing(ac,t);

NON-TERMINAL TASK
t_takeoff(ac:Aircraft, t:Travel)

Objectives:

level(location(flight(t))=CRUISING_LEVEL

NON-TERMINAL TASK
t_cruising(ac:Aircraft, t:Travel)

Objectives:

location(flight(t)) in f_approach_aera(location(flight(t)))

Methods:

journey_cruising{...},
circle_cruising{...}

MMETHOD
journey_cruising{...}

Preconditions:

ok(status(flight(t))) == true

Applicability conditions:

from_place(flight(t)) != to_place(flight(t))

Effects:

fuel_level(tank(aircraft_characteristics(ac))) − =f_consumption(flight(t)),
to_date(flight(t)) + = f_duration(flight(t)),
location(flight(t)) in f_approach_aera(location(flight(t)))

Control:

...

MMETHOD
circle_cruising{...}

Preconditions:

ok(status(flight(t))) == true

Applicability conditions:

from_place(flight(t)) == to_place(flight(t))

Effects:

location(flight(t)) in f_approach_aera(location(flight(t))),
fuel_level(tank(aircraft_characteristics(ac))) − =f_consumption(flight(t)),
to_date(flight(t)) + = f_duration(flight(t))

Control:

...

NON-TERMINAL TASK
t_landing(ac:Aircraft, t:Travel)

Objectives:

level(location(flight(t))) = GROUND_LEVEL

Methods:

gentle_landing{t_extend_gear, t_verify_gear, t_bring_down, t_roll, t_park}

DTL (Degraded Tasks Library)

In the DTL (Degraded Tasks Library), we consider the unfortunate case of landing gear failure:

NON-TERMINAL TASK
t_crash_landing(ac:Aircraft, t:Travel)

Objectives:

level(location(flight(t))) = GROUND_LEVEL
safe(passengers(t))

Methods:

landing_without_landing_gear{t_bring_into_play_security_procedure, t_choose_landing_area, t_bring_aircraft_down}

MMETHOD
landing_without_landing_gear{t_bring_into_play_security_procedure, t_choose_landing_area, t_bring_aircraft_down}

Preconditions:

fuel_level(tank(aircraft_characteristic(ac))) == LOW_LEVEL

Effects:

level(location(flight(t))) = GROUND_LEVEL,
safe(passengers(t)) = true,
damaged(status(aircraft_parameters(ac))) = true

Control:

t_bring_into_play_security_procedure(ac,t);
t_choose_landing_area(ac,t);
t_bring_aircraft_down(ac,t);

8

A Unifying Multimodel Taxonomy and Agent-Supported Multisimulation Strategy for Decision-Support

Levent Yilmaz[1] and Andreas Tolk[2]

[1] Department of Computer Science and Software Engineering, Auburn University, Auburn, AL, U.S.A. yilmaz@auburn.odu
[2] Frank Batten College of Engineering & Technology, Old Dominion University, Norfolk, VA 23529, U.S.A. atolk@odu.edu

Summary. Intelligent agent technology provides a promising basis to develop next generation tools and methods to assist decision-making. This chapter elaborates on the emergent requirements of decision support in light of recent advancements in decision science and presents a conceptual framework that serves as an agent-based architecture for decision-support. We argue that in most decision-making problems, the nature of the problem changes as the situation unfolds. Initial parameters, as well as scenarios can be irrelevant under emergent conditions. Relevant contingency decision-making models need to be identified and instantiated to continue exploration. In this paper, we suggest a multi-model framework that subsumes multiple submodels that together constitute the behavior of a complex multi-phased decision-making process. It has been argued that situation awareness is a critical component of experience-based decision-making style. Perception, understanding, and anticipation mechanisms are discussed as three major subsystems in realizing the situation awareness model.

8.1 Introduction

Decision science involves understanding cognitive decision processes, as well as methods and tools that assist decision-making (Davis et al. 2005). Significant amount of research has been conducted on decision theory and associated processes. This chapter focuses on how intelligent agent technology can provide basis for a unified synthesis of deductive, practical, and experience-based mechanisms to constitute a multi-level decision support system. In this context, logical, practical, and experience-based decision-making are analogous to rational choice model (von Neumeann and Morgenstern 1953), heuristics and biases (Tversky and Kahneman 1974), and naturalistic decision-making (Klein 1997).

Decision-making involves making tradeoffs among competing attributes or goals, analyzing complex situations within constraints of time and resources,

L. Yilmaz and A. Tolk: *A Unifying Multimodel Taxonomy and Agent-Supported Multisimulation Strategy for Decision-Support*, Studies in Computational Intelligence (SCI) **97**, 193–226 (2008)
www.springerlink.com

projecting into future state of the environment despite uncertainty, and making judgments, even if they are heuristic (Zachary 1998). The evolution of decision-making theory can be viewed as a steady withdrawal from the rational choice model to bounded rationality, and most recently to naturalistic decision-making (NDM) theory. While rational choice model (Parsons and Wooldridge 2002) involves the maximization or optimization of the expected utilities, bounded rationality emphasizes the constraints of time, resources, and cognitive capacities. Bounded rationality worldview involves the use of heuristics and biases (Tversky and Kahneman 1974) to capture cognitive shortcuts used in decision-making. Naturalistic decision-making, on the other hand, is based on the premise that humans assess situations by using prior experience. Zsambok (1997) argues that situation assessment and experience-based decision-making is more appropriate than option generation under conditions that involve uncertain and dynamic environments, shifting or competing goals, time stress, and ill-structured problems. Note that decision-making styles can shift between analytic, heuristic, and experience-based several times within a single problem (Hamm 1988). Furthermore, Hammond (1986) demonstrates that task features, such as complexity of the task structure, ambiguity, and form of representation, determine the decision-making style. More specifically,

1. In most realistic decision-making scenarios, the nature of the problem changes as the situation unfolds. Initial parameters, as well as scenarios can be irrelevant under emergent conditions.
2. Our knowledge about the decision problem being studied may not be captured by any single decision-making style. Instead, the available knowledge is viewed as being contained in the collection of all possible decision-making experiments that are plausible given what is known and what is learned.
3. Dealing with uncertainty is paramount to making decisions within the context of complex evolving phenomena. Dynamic adaptivity in decision-making styles is necessary to deal with emergent conditions or evolving decision-making process in a flexible manner.

Based on these observations and a recent recommendation (Davis et al. 2005), the contributions of this chapter are two-fold.

1. An agent-supported multisimulation approach that aims to simultaneously analyze multiple alternative Course of Actions (COAs), and, if necessary, update the scenario to deal with new phases of problem.
2. Delineation of the design considerations for the agent-based naturalistic decision-making.

Intelligent agents are proven to be useful in decision-making, especially within the context of game theory (Parsons and Wooldridge 2002) and mechanism design (Wooldridge 2002). Designing mechanisms refers to developing agent interaction protocols, called strategies, which satisfy desirable properties such as Pareto efficiency, stability, and social welfare maximization among a collection of agents. Power (Power 2002) describes how model-based decision support can be supported by simulation systems in general and agent-based simulation systems in particular. Tolk (2004) enumerates a comprehensive list of military decision-making functions for which agents can provide valuable support.

Proper simulation-based decision support methodologies that are consistent with the way experts use their experience to make decisions in field settings could improve modeling for Course of Action (COA) analysis. Each COA is simulated faster than real time, the results are collected, and COA analysis can be performed. Additional requirements for simulation systems when being used for this sort of analysis are summarized in (Tolk and Kunde 2003). Exploring the effectiveness of alternative COAs at the tactical and operational levels requires dynamic updating, branching, and simultaneous execution of simulations, potentially at different levels of resolution. We propose a strategy in integrating human-centered decision-making with multisimulation-based COA analysis. Three modes are identified:

1. Human-in-the-loop with naturalistic decision-making approach,
2. Agent-augmented naturalistic decision-making,
3. Agent-based naturalistic decision-making.

The first mode involves an operator that interacts with the simulation to choose alternative COAs based on situational awareness gathered from the results obtained during the simulation. The second mode aims to augment the decision-making process of the operator with intelligent agents that carry out routine tasks that pertain experience the situation in a changing context, reasoning about and diagnosing the situation to make recommendations for plausible COAs. In the third mode, intelligent agents replace the operator, and they perform the perception, understanding, and anticipation functions to model the situation awareness capabilities of the operator.

In many situations simulation specialists build a simulation and then conduct the special study and report their results to management. Evan and Olson (2002) discuss examples of how simulation has been used to support business and engineering decision-making. Their examples are prototypical for our findings: simulation systems without agents designed for reliable decision support are not universal tools, but special – and often expensive – means of operations research. The methods and technology described in this chapter help to make simulation systems flexible and reliable enough to become decision support systems.

The rest of the chapter is structured as follows. Section 8.2 presents the major decision-making styles, the decision-making process, and intelligent agents. Section 8.3 enumerates a set of requirements for next generation for intelligent simulation-based decision support systems based on the nature and types of emergent problems in various application domains. Section 8.4 introduces the macro-architecture for the proposed decision-support system.

We show how alternative decision styles can be supported within a multi-level view of the decision-making problem. Section 8.5 focuses on the design of situation-aware agents that are capable of augmenting humans to make experience-based decisions. It also presents selected research domains for the next generation of such systems. Section 8.6 presents a case study to substantiate the utility of the presented decision support approach. Finally, Sect. 8.7 concludes by discussing potential avenues of research and application.

8.2 Decision Science and Intelligent Agents

For our approach, we view decision-making as a cognitive reasoning process. The first subsection presents the characteristics of major decision-making styles. The second subsection overviews the process and its phases. The last subsection characterizes the role that intelligent agents can play to support each phase of the process (Fig. 8.1).

8.2.1 Decision-making Styles

Decision-making is viewed as a process that entails two distinct activities. The first one is to decide what state of affairs is desired and second how this state will be achieved. In modern decision science, there are mainly three decision-making styles.

- **Rational Choice Model (RCM):** This model of decision-making emerged in such diverse fields as economics, political science, management science, and operation research.

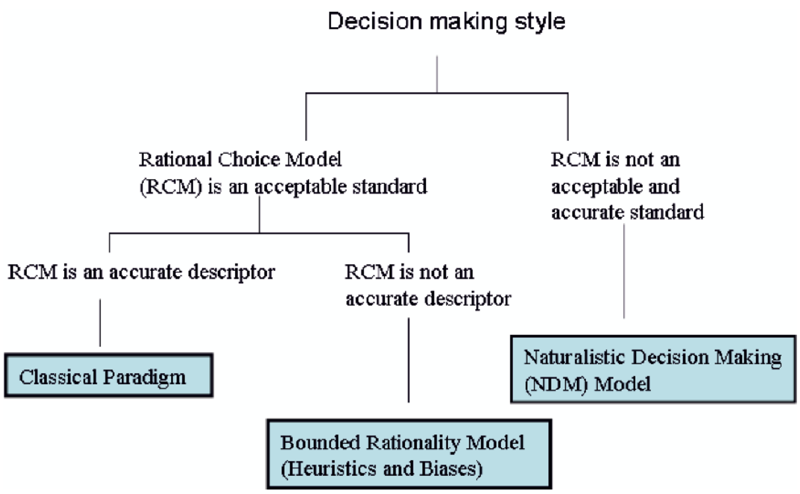


Fig. 8.1. Decision-making styles

von Neumeann and Morgenstern (1953) introduced the idea that rational choice should maximize expected subjective utility. From the perspective of game theory, this classic approach to decision analysis can be viewed as an analytical approach that optimizes the outcome of a decision. Building on the rationality principle, game theory has been applied to various problems (Geyer and van der Zouven 1998, Shubik 1964). However, evidence exists that classical game theory fails in cases where opponents have different value systems (Knight et al. 1991). Different types of game theories (e.g., sequential games, repeated games (Banks and Sundaram 1990, Leimar 1997)), differential games, evolutionary games, and hyper-games (Fraser and Hipel 1984), have been applied in the context of RCM.

- **Bounded Rationality (BR):** In making decisions, humans operate within a complex and often changing environment with limited cognitive capabilities, time, and other resources. Hence, decision-making is only rational within the bounds imposed on decision makers (Simon 1982).

  Tversky and Kahneman (1974) identified a number of heuristics and biases that humans use to make decisions. These studies aim to bring classical and analytic decision theorists into conformity with findings in cognitive psychology. The premise of bounded rationally is based on the observation that heuristics (Davis et al.2005) often yield cost-effective compared to classical methods in terms of time and mental effort. Furthermore, changes in the environment will cause the judgment to be obsolete.
- **Naturalistic Decision-making (NDM):** The empirical work of Gary Klein (1997) on expert behavior in high-pressure environments resulted in a new school of thought in decision-making. The NDM paradigm argues that people assess situations by using prior experience and knowledge.

  Furthermore, unlike RCM and BR decision-making styles, in NDM situation assessment is considered to be more important compared to option generation. Hence, the approach is to perform pattern matching to match observed problem facets to the mental model of the problem formed by the decision maker. Sokolowski (Sokolowski 2003) discusses the application of NDM for agent supported decision-making.

8.2.2 Intelligent Agents

In the context of this chapter, we use the definition of Ferber (1999), who defines software agents as entities that are capable of acting in purely software and/or mixed hardware/software environments

1. can communicate directly with other agents,
2. are driven by a set of goals, objectives and tendencies,
3. possess skills to offer services,
4. perceive its environment, and
5. can generate autonomous behavior that tends toward satisfying its objectives.

An overview of additional views is documented in Murch and Johnson (1998). Furthermore, we assume that the environment will be

- not-accessible (versus accessible),
- stochastic (versus deterministic),
- dynamic (versus static),
- sequential (versus episodic),
- and continuous (versus discrete)

to represent the environments specifies in the last section for realistic decision-making problems.

In this context, we understand agents as autonomous software modules with perception and social ability to perform goal-directed knowledge processing over time, on behalf of humans or other agents in software and physical environments. When agents operate in physical environments, they can be used in the implementation of intelligent machines and intelligent systems and can interact with their environment by sensors and effectors. The core knowledge processing abilities of agents include: reasoning, motivation, planning, and decision-making. The factors that may affect decision-making of agents, such as personality, emotions, and cultural backgrounds can also be embedded within agents. Additional abilities of agents are needed to increase their intelligence and trustworthiness. Abilities to make agents intelligent include anticipation (pro-activeness), understanding, learning, and communication in natural and body language. In this chapter, we advocate the use of (1) practical situation-aware agents that diagnose the situation via perception, understanding, and anticipation capabilities and (2) agents that facilitate simulation-based analysis of alternative COAs.

8.3 Requirements for Developing Computational Frameworks for Decision Support

Advances in decision science and the nature of problems being tackled impose new requirements on next generation decision-support systems.

8.3.1 Decision Styles and Problem Domain Characteristics

The nature of the decision style further imposes constraints on the decision-making models within a multi-model. Table 8.1 depicts the three main decision styles discussed in the earlier section along with the problem domain characteristics they target.

For instance, the RCM style provides an acceptable and accurate framework for problems in which actors, their preferences, utilities for actions, and the outcomes are well-defined. The problem is expected to be stable, and the number of options and players are small. Furthermore, the cognitive limitations of the decision maker and the lack of resources are not considered

Table 8.1. Features of decision-making styles

Decision-making style	Problem Domain Characteristics	Tool Design Features
Rational Choice Model	1- Well-defined problems 2- Low uncertainty 3- Stable environment 4- Small number of players and options 5- Time is not a parameter/factor	a- High-level design templates for various recurring problems b- Graphical interfaces for specifying utilities, actors, preferences, and outcomes
Bounded Rationality	1- Resource limitations (cognitive, computational etc.) 2- Time stress is a factor 3- Medium level certainty 4- Incomplete information about the environment	a- Models that encode heuristics and biases such as availability, representativeness, and anchoring and adjustment heuristics [1]
Naturalistic Decision Making	1- Ill-structured problems 2- Uncertain, dynamic environments 3- Shifting, ill-defined, competing goals 4- Action/feedback loops 5- Time stress and high stakes 6- Multiple players 7- Organizational goals and norms are factors (Zsambok 1997)	a- Perceiving situations in an environment b- Matching perceptions against learned experiences c- Understanding the overall situation via comprehension mechanisms d- Exploring possible outcomes by emulating mental simulation d- Anticipating future state(s) of the environment before making a decision

to inhibiting factors in decision-making. Having decision-making tools that enable formal specification of the structure of decision-making problem is feasible under these conditions.

Therefore, interactive tools that provide graphical facilities to capture options, preferences, utilities etc. can be useful. On the other hand, NDM decision-making style is introduced for problem domains that are ill-defined. The level of uncertainty in the environment leads to shifting and possibly competing goals. The characteristics of the domain are common in decision-making environments where there is a time stress, high stakes, and continuous action/feedback loops.

To support experts in making decisions in such environments, a decision-support system needs to provide facilities to augment pattern matching for situation recognition, understanding of the overall situation from the perceived disconnected elements, and make projection to potential future states. The projection phase simply involves tool support for mental simulation of the plausible actions.

8.3.2 Multisimulation in Support of Naturalistic Decision-making

Many real-world decision-making phenomena can not be modeled by one single model; rather, they require the use of a set of complementary decision-making models representing multiple perspectives that are able to describe the whole process possibly at different resolutions and phases when applied orchestrated (Bigelow and Davis 2003, Ören (1987, 1991, 2001), Zeigler et al. 2000, Yilmaz and Ören 2004). We distinguish contribution of multimodels and multisimulation that are dealt with in the following in more detail.

Multimodels

Models are purposeful abstractions of reality. Complex challenges require the use of several different views – or abstractions – to cover the full spectrum. This motivates the use of multimodels. While one big model is feasible, it is likely that this model would be as complicated as the real problem and the modeling would not result in any advantage. Several smaller models combined with each other overcome both shortcomings. Basic definitions and brief explanations of the envisioned multimodel types – as they are shown in Table 8.2 – follow here:

A multimodel is a modular model that subsumes multiple submodels that together constitute the behavior of a complex multi-phased decision-making process. A multimodel encapsulates several aspects of reality (i.e., submodels) in one model. For instance, conflict resolution problems discussed in (Yilmaz et al. 2006) emphasized the importance of dropping the notion of decision-making using a single conflict management procedure for the management and resolution of complex conflicts. Tolk (2004) discusses similar issues for agent mediated decision support in the military domain. The discussion on the use of multi-aspect, multi-stage, multi-resolution multimodels implies a certain type of conflict dynamics; that is, a set of stages in the process associated with proper conflict management procedures for each stage.

Note however, that as a situation unfolds, the parameters of the decision and payoff matrices, the state space of the problem, the attitudes, and preferences may change. Therefore, the time path of a decision-making process should map onto a time path of decision-making styles embedded within the models. Critical questions that need to be answered include the issues pertaining to the mechanism by which decision-making styles are selected, when and how shifts occur in updating multimodels, and to whom the judgment to determine the shift should be given. In single aspect models only one aspect of reality can exist at a given time (to be represented by an appropriate submodel) and transitions can occur from one submodel to another one under monitored conditions. Special cases of multimodel formalism are the metamorphic model and the evolutionary model.

A metamorphic model has a fixed number of submodels with a predetermined transition order between the submodels. The transition conditions can include the processes of the metamorphosis.

Table 8.2. Synopsis of the envisioned mulimodel (MM) formalism

Based on	Additional Criteria				Type of Multimodel (MM)
Structure of submodels	Number of submodels active at a given time			Only one	Single Aspect MM (sequential MM)
				Two or more	Multi-aspect MM
	Variability of structure (variability of number of submodels)	Static			Static-structure MM
		Dynamic	Number of submodels	Extensible	Extensible MM
		(Dynamic structure MM)		Depends on model's stage	Multistage MM
		(Variable structure MM)	Alternations of submodels	No	Non-mutational MM
				Yes	Mutational MM
					Evolutionary MM
Behavior (activation) of submodels	Nature of knowledge to activate the submodels	Constraint-driven			Constraint-driven MM (Adaptive MM)
		Pattern-directed MM (Metamorphic MM)	Submodel selection is cyclic	No	Acyclic MM
				Yes	Cyclic MM
		Goal-directed			Goal-directed MM (Exploratory MM)
	Location of knowledge to activate the submodels	Within the MM (internal activation of submodels)			Active MM (Internally activated MM)
		Outside the MM (External activation of submodels)			Passive MM (Externally activated MM)

An evolutionary model can have several submodels. The number of submodels at the beginning may be fixed or unknown. Subsequent submodels are variant models of their predecessors. The transitions from a submodel to another one can be achieved as rule-based, pattern-directed, or goal-directed activities. Evolution, being an irreversible change in an open system, is important in the study of decision-making. Mutations, pathological or not, –including social mutations– can be modeled as evolutionary models.

A multi-aspect model consists of several submodels where two or more submodels can be active at a given time. Since each submodel can represent an aspect of reality, several aspects –even contradictory ones– can be represented at the same time. The multi-aspect modeling methodology appears to be very promising to encapsulate several aspects of phenomena and their mutual influences. In a multi-aspect model, submodel(s) inactive at a given time are latent or dormant submodels. In decision-making, for instance, anticipatory study of the effects of latent submodels may deter later catastrophes.

A multistage model is a set of variable number of submodels that can be used to represent reality at different emerging stages of a system. In conventional decision-making studies, one model is used for the duration of the lifespan of a system. However, in social systems, the fluidity of the situation may necessitate exploring with more than one model at every emerging stage of the analysis.

As shown in Table 8.2, there are various design decisions in multimodel design. Alternative names are given in parentheses.

Based on the completeness of submodels, there are two cases: (1) one can either know all the submodels at the beginning i.e., at modeling stage, or (2) there can be emergent conditions where the need for additional submodels.

Based on the number of active submodels, one needs to consider two cases: (1) only one submodel is active at a given time or (2) two or more submodels are active at a given time. Simultaneous existence of two or more model components would facilitate simulation of multiple aspects of the phenomena under study.

Based on the location of information necessary for the activation of submodels there are two cases: the necessary information can be (1) within the submodels or (2) it can be external to submodels.

The transitions between submodels can be goal-directed (goal directing the submodel transition rule and goal-directed submodel transition mechanism should be specified) or pattern-directed. Natures of information necessary for the activation of submodel(s) entail the selection conditions of a submodel.

Pattern-directed activation entails a meta-pattern to guide (1) selection of known submodels and (2) request of new submodels corresponding to an interruption of the decision-making process using a human-in-the-loop mechanism.

Multisimulation

We define multisimulation as a simulation of several aspects of reality in a study. It includes simulation with multimodels, simulation with multi-aspect models, and simulation with multistage models. Simulation with multimodels allows computational experimentation with several aspects of reality; however, each aspect and the transition from one aspect to another one are considered separately. (In special cases, multimodels can be metamorphic models or evolutionary models). Simulation with multi-aspect models (or multi-aspect simulation) allows computational experimentation with more than one aspect of reality simultaneously. This type of multisimulation is a novel way to perceive and experiment with several aspects of reality as well as exploring conditions affecting transitions. While exploring the transitions, one can also analyze the effects of encouraging and hindering transition conditions. Simulation with multistage models allows branching of a simulation study into several simulation studies; each branch allowing to experiment with a new model under similar or novel scenarios.

In our approach, there can be multiple strategy components that are qualified at the time of decision-making. Each different strategy component characterizes a distinct aspect. Multisimulation can be used to branch out multiple simulations, where each simulation uses a specific component configured with an exclusively selected strategy component. Similarly, multiple distinct stages of the problem can be qualified at a given point in time during the simulation by virtue of the evaluation of an updating constraint. In such a case multisimulation enables branching multiple distinct simulations each one which generates the behavior of distinct plausible stage within the problem domain.

Multisimulation with multimodels, multi-aspect models or multistage models needs mechanisms to decide when and under what conditions to replace existing models with a successor or alternative.

Staging considers branching to other simulation studies in response to a scenario or a phase change during experimentation. Graphs of model families facilitate derivation of feasible sequence of models that can be invoked or staged. More specifically, a graph of model families is used to specify alternative staging decisions. Each node in the graph depicts a model, whereas edges denote transition or switching from one model to another. Figure 8.2 depicts the components of the abstract architecture of a possible multisimulation engine.

A meta-simulator is a scheduler that generates staged composition of models by traversing the model stage graph and coordinates their simulation and staging within distinct simulation frames. Each frame simulates a distinct subset of models derived from the model stage graph. Note however, that not all staged compositions are feasible or useful. Hence, the meta-simulator needs to consult with the model recommender before model staging to determine if emergent trigger or transition condition in the simulation is consistent with

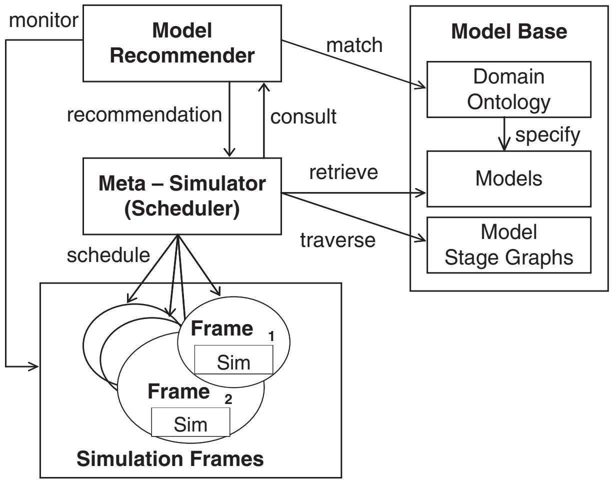


Fig. 8.2. Abstract components of the multisimulation engine

the precondition of the model to be staged. More than one model in a family can qualify for staging; in such cases separate simulation frames need to be instantiated to accommodate and explore plausible scenarios. Given a collection of models (or more generally, a family of models), a stage graph can be generated automatically by an optimistic approach that connects every available node (model) to every other node within the domain of problem. The edges in a model stage graph denote plausible transitions between models as the problem shifts from one stage to another. One can consider each model as a separate conflict management protocol (i.e., compromise over actions, compromise over outcomes, negotiation, and mediation) or a phase in the conflict process (i.e., escalation, resolution), where a phase (i.e., resolution) can constitute alternative models (i.e., mediation, negotiation, third-party intervention).

The subsets of staged models can be identified by traversing and enumerating the graph in some order (i.e., depth-first). Infeasible paths may be due to an unreachable node, or it may result due to conflicts between the transition condition and precondition of the target model. Infeasible paths due to incompatible sequences of models are common. Each edge (say from n_i to n_j) indicates that there is some legitimate solution that includes n_i followed by n_j; yet, it does not imply that every solution containing n_i followed by n_j is legitimate. As argued above, each model in a family of models is associated with a precondition. A precondition denotes the conditions required for a model to be instantiated. Hence, the feasibility of staging a successor model depends on the satisfiability of its precondition (relevance) by the condition of the transition and the post-condition of the predecessor model. As a result, not all enumerated staged sequences of model components are feasible.

Model recommendation in multisimulation can simply be considered as the exploration of the model staging space that can be computed by a reachability analysis of the graph. There are two modes for the usage: (1) offline enumeration of paths using the graph and performing a staged simulation of each model in sequence one after the other, unless a model staging operation becomes infeasible due to conflict between the transition condition and the precondition of the successor model and (2) run-time generation of potential feasible paths as the simulation unfolds. In both cases, an online model recommender plays a key role to qualify a successor model. The first case requires derivation of sequence of models using a traversal algorithm. The edges relate families of models. Therefore, the actual concrete models, the preconditions of which satisfy the transition condition need to be qualified, since transition to some of these model components may be infeasible due to conflict between a candidate model and inferred situation. Identifying such infeasible sequences is computationally intractable; otherwise, it would have been possible to determine if the conjunction of two predicates is a tautology by using a polynomial time algorithm.

Experience in the component-based simulation paradigm, however, indicates that for most model components preconditions are simple. Hence, it is possible to eliminate some models that violate the transition condition. For the remaining possible transitions it is possible to select one of the three strategies: (1) omit all difficult qualification conditions, (2) decide on an edge-by-edge basis which specific models of a model family to include, and (3) include all difficult edges. Omitting all difficult associations between transitions and model preconditions is conservative. This strategy excludes all infeasible models. The cost is the exclusion of some feasible edges. Hand-selecting those associations between transition conditions and models facilitate inclusion of feasible models. Nonetheless, the costs involved with this level of accuracy are the potential human-error and effort needed to filter out infeasible models. Choosing to include all difficult associations is liberal, in that it ensures inclusion of all feasible models. The cost is the inclusion of some infeasible models, hence the inclusion of some undesirable staged compositions that enforce models to be simulated even when their qualification conditions are violated. Nevertheless, it is possible to screen out such models using an online model recommender.

The second more ambitious yet flexible approach is to delay the enumeration process until a model is qualified at run-time. Runtime generation of feasible staging using the graph of model families requires monitoring and evaluation of transition conditions as the simulation unfolds. A planning layer connected to simulator would be capable of identifying, qualifying, and, if necessary, selecting and instantiating a model based on the specified preferences and options. Furthermore, in the case of an impasse or lack of knowledge on preferences among qualifying model switch strategies, a planning layer can guide exploring alternative contexts (games) in some order. The metascheduler follows the recommendations made by the planner to instantiate distinct simulation frames.

Candidate models and associated simulations are maintained by focus points. A focus point manages branch points in the simulation frame stack. Suppose that a goal instance (i.e., stage transition condition) is at the top of the stack. If only a single model qualifies for exploration, then it is pushed onto the stack. Yet, if more than one model matches the condition, a simulation focus point is generated to manage newly created simulation branching (discontinuity) points. Each one of these simulation focus points has his own context. When a path is exhausted, the closest focus point selects the next available model to instantiate the simulation frame or return to the context that generated the focus point. As simulation games are explored, a network of focus points is generated. Determining which focus point should be active at any given time is the responsibility of the meta-scheduler. When more than one model is qualified, then scheduler needs to decide which one to instantiate. Control rules can inform its decision. Three steps involve in deploying a new simulation frame in such cases: matching, activation, and preference. The matching step should both syntactically and semantically satisfy the request. The activation step involves running a dynamic set of rules that further test the applicability of models with respect to contextual constraints. Finally, the preference steps involve running a different set of rules to impose an activation ordering among the active frames.

8.4 Agent-Based Intelligent Decision Support – A Unifying Framework

We present a unified exploratory multisimulation technology, which suggests a simulation world-view shift. After evaluating general observations, we will focus on aspects of situation awareness and experience-based reasoning.

8.4.1 Architectural Constraints for a Unifying Framework

Experimentation with exploratory multisimulation contrasts sharply with establishing a base-case model and scenario to perform sensitivity and factor analysis, where the user is interested in understanding the variance of predictions under priory selected configurations. Exploration involves performing computational experiments under uncertainty to gain intuition about possible outcomes, if decisions on using certain models based on emergent conditions are true. The premise of exploratory multisimulation is based on the view that the results of a simulation are not viewed as a prediction of what we would expect to occur, but rather the results of a computational experiment. By making recommendations for staging and branching to alternative models as well as scenarios, dynamic simulation update mechanisms enable exploratory multisimulation.

As exploration is based on a number of such recommendations, our knowledge about the problem being studied cannot be captured by any single model, scenario, or experiment. Instead, the domain knowledge needs to be viewed as being contained in the collection of possible modeling experiments and ensemble of models that are plausible given what is known or learned during the simulation experiment. Multisimulation subsumes multi-resolution simulation, where entities are capable of simultaneously operating at different levels, while maintaining consistency at each level of abstraction.

Embedding such a decision-centered simulation methodology into operational systems is a significant challenge. Operational necessity and integration concepts are discussed have been discussed among others by Daly and Tolk (2003). In decision-making situations, operators should be able to identify and investigate the impact of COAs to evaluate the effectiveness of decisions. To this end, a decision support system based on exploratory multisimulation technology that will operate within the framework of NDM. NDM is emerging as a field of research, providing a descriptive view of how people behave in dynamic, uncertain, and often fast-paced environments. This model focuses on experienced agents, working in complex, uncertain conditions, who face personal consequences for their actions. Figure 8.3 depicts the organizational layout of the components that constitute the solution. In the following sections we will clearly identify the technologies, (basic, applied research, or

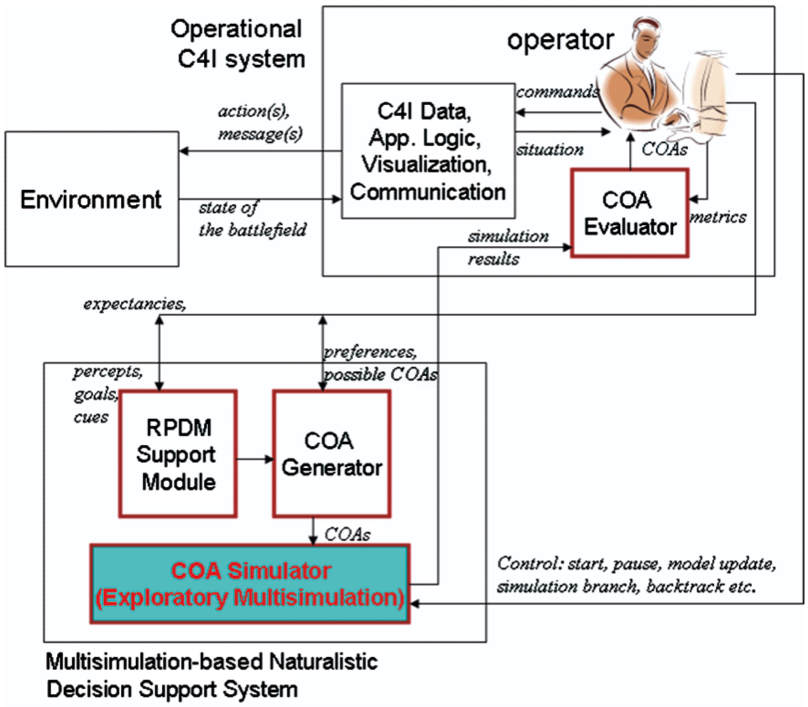


Fig. 8.3. Architecture of the decision-support system

exploratory development) forming the proposed solution. The premise of the approach is that decision makers (i.e., operators) need tools to augment their decision-making process. Such a decision support tool, however, needs to be consistent with how experts use their experience to make decisions in operational settings. To this end, we choose an NDM framework, which provides a descriptive view of how people behave in dynamic, uncertain, and often fast paced environments. NDM focuses on experienced agents, working in complex, uncertain conditions, who face personal consequences for their actions (Zsambok 1997). Development and insertion of this technology into operational systems forms the basis of the technical objective. The novel aspects of the approach are based on the following technologies.

- Exploratory multisimulation that realize the mental simulation component of Recognition-Primed Decision (RPD). Dynamic model and simulation updating is a novel strategy that enable evaluating multiple COAs via simulation branching.
- A computational model for situation-aware RPD, which is a special case of NDM, and
- Agent-supported COA generation based on practical agent reasoning technology.

The operational C4I system shown in Fig. 8.3 embodies a multisimulation-based decision support subsystem that aims to evaluate various COAs on behalf of the operator. The operator interprets the situation in consultation with the computational RPD model to generate valid and accurate percepts based on his experience. RPD component provides a computational mechanism for situation recognition and pattern recognition. The output of the RPD Making (RPDM) module is a set of goals, expectancies, and clues. This output is evaluated by the operator to generate a set of preferences and/or action(s) to be carried out by the simulation component of the decision support system. The preferences and actions are used by the COA generator component that deploys an agent–based planning algorithm to generate a set of plans. These plans are then simulated by the exploratory multisimulation engine. The simulation results are then evaluated interactively by the operator using the COA filter that uses the provided performance metrics.

8.4.2 Situation Awareness and Experience-based Reasoning

The decision support system is designed to support three modes of operation – operator-driven, agent-augmented, and agent-supported multisimulation.

Mode 1: Operator-driven Multisimulation

The first mode is the operator acting on his/her own interpretation of the situation to devise COAs. The strategy is as follows:

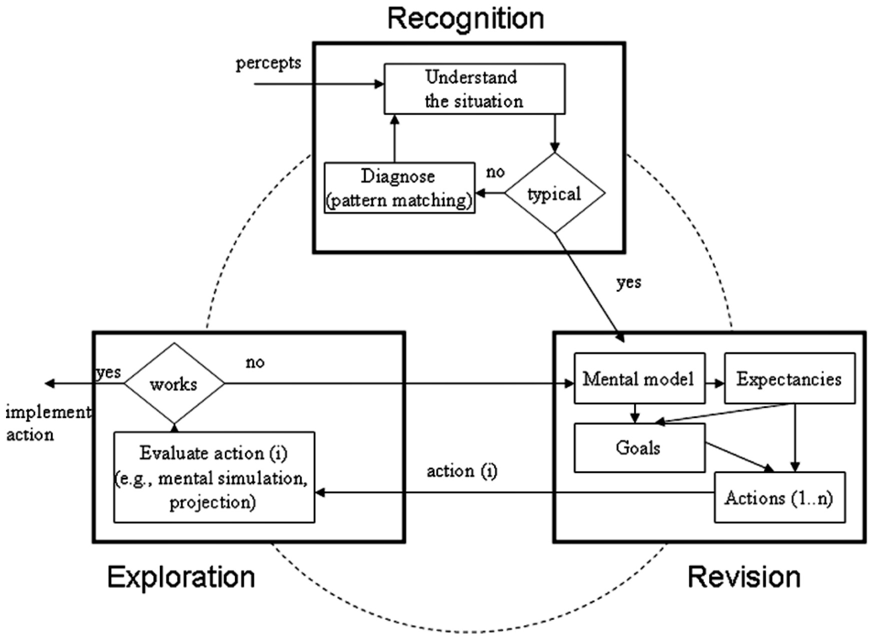


Fig. 8.4. Experience-based reasoning

1. Observe the C4I system
2. Perceive and understand the situation
3. Anticipate/project future status
4. Decide on plausible COAs
5. Update the simulation model to predict outcomes under alternative COAs.

The first three steps in the above strategy refer to diagnosis of the situation. The diagnosis activity is initiated in response to uncertainty about the nature of the situation. The life cycle for experience-based decision-making of the operator involves three main stages as shown in Fig. 8.4: the recognition, revision, and exploration phases. The architecture embodies an extended version of the RPD model (Klein 1997). The model, which is based on Recognition Primed Decision Model, is an example of NDM, and it attempts to emulate what people actually do under conditions of time pressure, ambiguous information, and changing conditions. According the architecture, the sensory input is processed by the experience the situation component to perceive the elements of the situation. If the situation is prototypical, the NDM submodel instantiates a skeleton mental model, from which expectancies and goals can be derived. Simple if-then rules can be used to derive plausible actions based on goal-action pairs. These goal-action pairs are based on prior experience, and they are encoded within the mental model. If the observed situation and perceived inputs are not categorized to be prototypical, then a diagnosis (i.e., pattern matching) procedure that synthesizes the features of the percepts to causal factors is enacted to facilitate comprehending the situation until a prototypical or analog case is identified.

The exploration phase of the life cycle requires evaluating the selected action. Humans often perform mental simulation of the possible outcomes if and when the decision is implemented. In our system, the evaluation is performed via multisimulation. If the action is found to be irrelevant to the goal as a result of the projection or mental simulation, the mental model is further revised to either update the goal or identify a different action. The challenge in this mode is in providing a front-end interface to multisimulation to pause, update, reconfigure, and restart the simulation with the new parameters, models, and even scenarios. In this mode, the operator will browse through the available COA in the library or query based on the perceived situation. The recognition and revision phases are manual, whereas evaluation is supported by multisimulation. However, the update operations over the multisimulation are still manual.

Mode 2: Agent-Augmented Multisimulation

In this mode, the operator is active in perceiving the situation, understanding it, and projecting the status for decision-making. However, unlike the operator-centered mode, intelligent agents are responsible for dynamically updating the model. Our design strategy for enabling this operation is based on an ontology-driven approach that provides introspective access to dynamic object patterns. More specifically, the multisimulation provides the facilities that

1. establish a self-representation of the system using dynamic object pattern ontologies,
2. offer means by which this representation can be updated, and
3. assure that the manipulations to the self-representation influence the behavior of the system.

In effect, the system's self-representation is connected to the behavior of the actual application. Hence, the structure of an application is divided into two components: (1) system level and (2) meta-system level. The system level includes the stable components of the model, application level software objects, and the structural and behavioral dependencies between the components it includes. The meta-system level includes components that are subject to change, and the ontology is based on the dynamic object pattern. The meta-system level provides an interface to facilitate configuring or updating the ontology that subsequently drive the simulation. The meta-system level provides three categories of functions:

- **Reflection:** System level can access information about the system via facilitator agents associated with the system. This information can then be used to guide the behavior of the system.

- **Introspection:** System level can access and update the parameters of existing meta-simulation entities. This enables seamless and transparent update of the behavior of the system, since the behavior is influenced by the meta-system entities.
- **Intercession:** System level can change, exchange, insert, or remove meta-system entities and their connections to the system level. This feature enables dynamically including or inserting new components into the application at run-time

Mode 3: Agent-Supported Multisimulation

This mode of the decision support system involves the exclusive use of agents, and there is no operator in the loop. That is, the recognition, revision, and exploration components of the decision-making lifecycle are supported by intelligent agents. This mode requires further research on developing means to facilitate situational awareness for implementing the recognition and revision components of the decision-making life cycle. The recognition, revision, and exploration phases of the situation awareness layer, shown in Fig. 8.3, suggest three main functional areas that revolve around a mental model of the problem domain. More specifically, a well-defined mental model provides

1. knowledge about the concepts, attributes, associations, and constraints that pertain to the application domain,
2. a mechanism that facilitates integration of domain elements to form an understanding of the situation, and
3. a mechanism to project to a future state of the environment given the current state, selected action, and the knowledge about the dynamics of the environment.

Endsley (1995) defines situation awareness as the perception of elements in a particular environment within time and space, the comprehension of their meaning and the projection of their status in the near future.

8.5 Considerations for the Design of the Situation Awareness Subsystem

Situation awareness, as depicted here, provides a set of mechanisms that enable attention to cues in the environment, expectancies regarding future states. In realistic settings, establishing an ongoing awareness and understanding of important situation components pose the major task of the decision maker. Therefore, situation awareness is the primary is basis of the decision-making process in experience-based decision-making process (i.e., NDM).

Situation awareness, the mechanisms of which are shown in Fig. 8.5, is an important cognitive skill that is essential for expert performance in any field

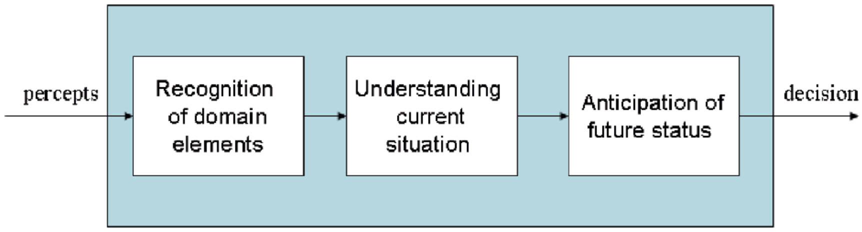


Fig. 8.5. Mechanisms for situation awareness

involving complexity, dynamism, uncertainty, and risk. The percepts are the interfaces to the environment; through them, the environment is perceived. The failure to perceive a situation correctly may lead to faulty understanding. Ultimately, this misunderstanding may degrade an individual's ability to predict future states and engage in effective decision-making (Gaba and Howard 1995). It is therefore an essential part of the NDM.

8.5.1 Perception

The way we perceive reality affects our feelings, decisions, and actions. Since Plato's allegory of the cave explained in Book 7 of "The Republic," it is well known that perception is very important (Bloom 1968). Wikipedia encyclopedia explains philosophy of perception as follows:

> "The philosophy of perception concerns how mental processes and symbols depend on the world internal and external to the perceiver. Our perception of the external world begins with the senses, which lead us to generate empirical concepts representing the world around us, within a mental framework relating new concepts to preexisting ones. Because perception leads to an individual's impression of the world, its study may be important for those interested in better understanding communication, self, id, ego –even reality." (Wikipedia (Phi-Per) 2004)

There are two types of perception, i.e., external and internal perceptions. Philosophy of perception is concerned with external or sensory perception.

> "External or sensory perception, tells us about the world outside our bodies. Using our senses of sight, hearing, touch, smell, and taste, we discover colors, sounds, textures, etc., of the world at large.
> Internal perception tells us what's going on in our bodies. We can sense where our limbs are, whether we're sitting or standing; we can also sense whether we are hungry, or tired, and so forth." (Wikipedia (Phi-Per) 2004)

Both types of perceptions can involve thought processes. Introspection is the detailed mental self-examination of feelings, thoughts, and motives.

Table 8.3. Categories of perception

	Current images of Past or current state	Future state
Others (people and/or events)	Perceived image of others and events	Behavioral anticipation of others and events
Self (decision maker(s), supporters, followers, and/or events related with one's own side)	Perceived image of self and/or events related with one's own side	Behavioral anticipation of self and/or events related with one's own side

> "In psychology and the cognitive sciences, perception is the process of acquiring, interpreting, selecting, and organizing sensory information. Methods of studying perception range from essentially biological or physiological approaches, through psychological approaches to the often abstract 'thought-experiments' of mental philosophy." (Wikipedia (Phi-Per) 2004)

A categorization of perception is given in Table 8.3. Perception of an entity at a time t gives an image of it at that time. At time t, we can refer to the perception as the current perception (or current image), if there is only one perception.

However, at a time t, based on the perspective, there may be different interpretations of an entity, hence several perceptions. From now on, for the sake of simplicity, unless it is specified otherwise, current perception (or current image) is considered to be unique. Current image can refer to external perceptions; hence it can be about others (people, groups, nations, events, facts, etc.). When current image refers to internal perceptions, then it is about the self (or own group of decision makers, supporters, followers; and/or events related with one's own side.) Current images may refer to past, current, or future states. There can be several current images, at different times $t_i,\ i = 1,\ 2,\ 3, \ldots,\ n$; until future becomes current.

This is similar to for example, seven day meteorological forecasts. At each day, there can be a forecast of a certain day until that day. And due to the variability of meteorological conditions, the forecasts may be different. When that specific day occurs, what we experience is the current image of the current state. If we are interested to interpret past events, current images of a certain past may be defined. However, there can be several images of a certain past based on the points of views of the people involved. Current images of (past, current, or future states) can reflect possibly different interpretations of the current perceptions. Hence, especially in a conflict situation, the opponents may even have antagonistic interpretations of the same situation. Furthermore, emotions such as anger affect the disposition of the decision makers.

8.5.2 Understanding

Understanding or comprehension of the situation is based on synthesizing the perceived disjoint elements to form a coherent representation of the entity, the elements of which are observed. For instance, the tactical commander of a military unit needs to comprehend that the appearance of enemy aligned in a specific pattern and in a particular location depicts certain specific objectives. Augmenting decision makers by providing capabilities that integrate perceived domain elements to facilitate comprehension of the situation requires taking the following design consideration. In the study of natural phenomena, the role of simulation is often cited as "to gain insight" which is another way of expressing "to understand." Understanding is one of the important philosophical topics. From a pragmatic point of view, it has a broad application potential in many computerized studies including program understanding, machine vision, fault detection based on machine vision as well as situation assessment. Therefore, systematic studies of the elements, structures, architectures, and scope of applications of computerized understanding systems as well as the characteristics of the results (or products) of understanding processes are warranted.

Dictionary definitions of "to understand" include the following:

- to seize the meaning of,
- to accept as a fact, believe,
- to be thoroughly acquainted with,
- to form a reasoned judgment concerning something,
- to have the power of seizing meanings, forming reasoned
- judgments,
- to appreciate and sympathize with, to tolerate,
- to possess a passive knowledge of a language

The following is a good starting point for the specification of the scope of machine understanding:

> "... if a system knows about X, a class of objects or relations on objects, it is able to use an (internal) representation of the class in at least the following ways: receive information about the class, generate elements in the class, recognize members of the class and discriminate them from other class members, answer questions about the class, and take into account information about changes in the class members." (Zeigler 1986)

From this point of view, knowing and computerized understanding can be taken as synonyms. However, one should remark here that knowing (something, somebody, some event, etc.) refers to the result of the process of acquiring knowledge and not the knowledge processing activity required to know. A system A can understand an entity B if three conditions are satisfied (see Fig. 8.6):

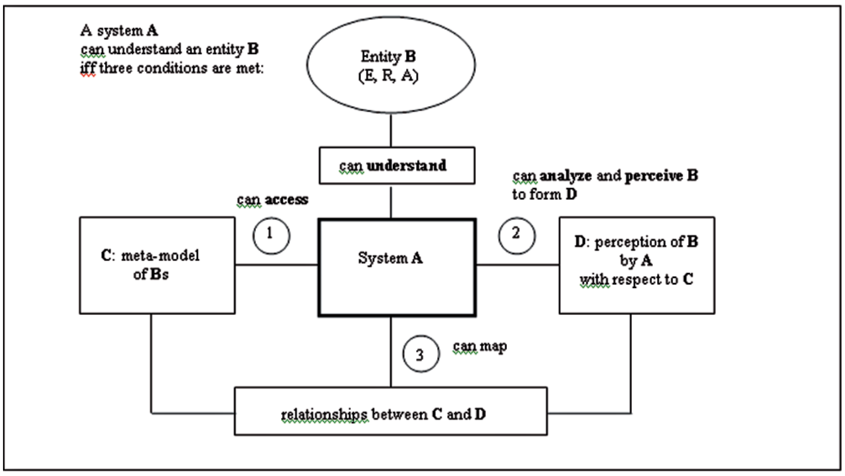


Fig. 8.6. Elements of an understanding system

1. A can access C, a meta-model of Bs. (C is the knowledge of A about Bs.)
2. A can analyze and perceive B to generate D. (D is a perception of B by A with respect to C.)
3. A can map relationships between C and D.

Therefore, an understanding system needs to have the following three basic elements: a meta-model of the entities to be understood, a perception element and an analyzer and a comparator to map a perception of an entity to be understood with the meta-model.

8.5.3 Role of Anticipation in Decision-Making

Anticipation is an important characteristic of intelligence. Pro-active behavior requires anticipatory abilities. Without anticipation a system can only be reactive; but a dead frog can also be reactive. A seminal work on anticipatory systems is the one written by Rosen (1985). A brief introduction to and serious concerns about anticipation follows:

> "Strictly speaking, an anticipatory system is one in which present change of state depends upon future circumstances, rather than merely on the present or past. As such, anticipation has routinely been excluded from any kind of systematic study, on the grounds that it violates the causal foundation on which all of theoretical science must rest, and on the grounds that it introduces a telic element which is scientifically unacceptable. Nevertheless, biology is replete with situations in which organisms can generate and maintain internal predictive models of themselves and their environments, and utilize the predictions of these models about the future for purpose of control in the

> present. Many of the unique properties of organisms can really be understood only if these internal models are taken into account. Thus, the concept of a system with an internal predictive model seemed to offer a way to study anticipatory systems in a scientifically rigorous way." (Rosen 1985)

A systematic review of 12 definitions of anticipation is available from Berkley Initiative in Soft-Computing, Special Interest Group (BISC-SIG) in Anticipatory Systems with the following warning:

> "The following 12 definitions, or descriptions, of anticipation should be understood as working hypotheses. It is hoped and expected that the knowledge community of those interested in anticipation will eventually refine these definitions and suggest new ones in order to facilitate a better understanding of what anticipation is and its importance for the survival of living systems."(BISC-SIG 2004)

An important aspect from the point of view of BISC-SIG is the emphasis on soft computing requirements in anticipation. Perception ability is a required characteristic of agents. Hence, they can be designed to perceive current state of self and others. They can also be designed to create current images of future states. An anticipatory system is a system whose next state depends on its current state as well as the current images of its future states. This definition is a radical departure from the original definition given by Rosen (1985): *"An anticipatory system is a system determined by a future state. The cause lies in the future."* Nonetheless, our definition is in line with the following definition also given by Rosen:

> "An anticipatory system is a system containing a predictive model of itself and/or of its environment that allows it to change state at an instant in accord with the model's predictions pertaining to a later instant." (Rosen 1985)

However, we would like to stress the distinction on dependency of next states on current images of future states rather than the future value of the states.

Perception requires mechanisms that enable interpretive capabilities. Perception invariably involves sensory qualities, and introspection entails accessing sensations and perceptions the agent would introspect. Perceptions are derived as a result of interpretation of sensory inputs within the context of the current world and agent's self model. The prototype inference, orientation accounting, and situational classification mechanisms (Sallach 2003). could be used to realize the interpretation capabilities of an agent. The interpretation process results in perceptions. An anticipatory agent needs to deliberate upon perceptions through introspection and reflection to anticipate.

Introspection is deliberate and attentive because higher-order intentional states are themselves attentive and deliberate. An introspective agent should have access mechanisms to its internal representation, operations, behavioral potentials, and beliefs about its context. Reflection uses the introspective

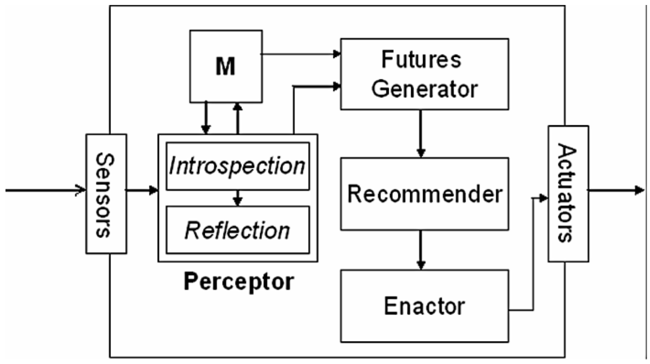


Fig. 8.7. Basic components for anticipatory agents

mechanisms to deliberate its situation in relation to the embedding environmental context. These features collectively result in anticipation capabilities that orient and situate an agent for accurate future projections. Figure 8.7 presents interpretation and introspection as critical components within the micro-architecture of an anticipatory agent. A computationally anticipatory agent needs to incorporate interpretation facilities as a precursor to (1) comprehend and draw accurate inferences about the world, (2) have social pragmatism by considering the likely responses of others in its context in response to a communication or act, and (3) have situational definition [40] as a direct input to action recommendation. An anticipatory agent uses a domain model M, as the internal representation of the environment and agent's self in order to project to the future. The model and the anticipation that results from the introspection and reflection processes are used to derive a number of realities by the futures generator. The generator is a function that maps environmental parameters and past vector of states onto a set of future states of the environment.

Naturally, an inductive process would be used to realize the function, as the generation of future plausible realities (environmental contexts) results in a set of new models that vary from each other based on assumptions on different plausible events or possible interactions between the environment and the agent itself. This perspective is consistent with the definition of anticipation process that is given in (BISC-SIG 2004). According to the definition, anticipation (1) is a realization within the domain of possibilities and/or (2) involves the generation of a multitude of dynamic models and the resolution of their conflict. As such, the recommender subsystem is responsible for evaluating alternative anticipated models and to decide on choosing a specific strategy based on the goals and motivations of the agent. Next, a recommender system should select a desirable future state upon which the agent would make decisions and react using its enactor component.

Developing anticipatory agents with run-time recommenders is difficult, because interpretation of emergent conditions requires mining the state of the simulation to recognize situations within the domain theory (schema) of an application. That is plausible and desirable future states need to be qualified based on the motives and goals of the agents. Learning takes place as recommendations are made. Adaptive models that assume certain discernible patterns in the recommendations may be used to discover situations and associated relevant models so as to reinforce qualification of specific future states based on previous experience. Various domain specific representational issues and inadequacies make this very difficult for particular applications. One form of representational inadequacy pertains to intrinsic difficulty of determining (and utilizing) the features that are potentially relevant for model selection. Another form of representational inadequacy involves on deciding the right level of detail. A major difference between traditional deliberative agents and an anticipatory agent is that an anticipatory agent makes guesses about the future state of the environment to guide its behavior, whereas conventional deliberative agents make their decisions based on the observed conditions within the current context.

8.5.4 Additional Research Domains

So far, we focused on decision makers as individuals. In the netted organizations supporting complex systems of today, this is no longer the rule. What is needed are good models for shared situation awareness, which in turn request good communication models between decision makers, representing agents, or supporting agents. Tolk and Gaskins (2006) published some tentative results in the light of the development of the Global Information Grid, a highly interconnected web-based infrastructure to support operations in the defense and security domains.

Recent work shows the challenge of building human behavior models in complex and cognitive domains. Cannon-Bowers et al. (1993) introduced the concept of shared mental models to describe the fluid, implicit interaction often observed in successful teams. Teams must predict and cope with task difficulty and change by altering their strategies. Shared mental models are the mechanisms that help teams make sense of situations and facilitate coordinated team performance and decision-making. Team members typically do not share a single mental model. Rather, there are likely multiple mental models co-existing among team members. Such shared mental models are characterized by a variety of factors including the characteristics of the team, the nature of the task, the type of equipment, and the interaction among the team members. However, these factors are generally categorized as either task work or teamwork mental models. Task work mental models include the understanding of activities and action sequences of the task, whereas teamwork mental models refer to the understanding of communication needs, compensatory behaviors, performance monitoring, and internal coordination strategies of the team. It has been shown that shared mental models relate positively to team processes, in particular decision-making, as well as performance.

Furthermore, team processes were found to fully mediate the relationship between shared mental models and performance. Although empirical support is limited, emerging findings suggest that appropriate team mental models have positive effects on team processes and effectiveness. Such findings suggest that the development of shared mental models is a promising leverage point for distributed learning techniques aimed improving team effectiveness. How these research results can be incorporated into agents in the light of these findings, is the subject of current research.

One of the most critical aspects of distributed decision-making environments is the role of information transfer between team members, i.e., communication. Researchers have studied the communication process for many years, and have constructed models to depict that process. Since Shannon and Weaver (1949) proposed one of the earliest models of the communication process based on telephone communications in 1949, research has focused on how information is transmitted and what are disturbing factors, such as noise or external events. A critical component of the model is noise, which may serve to confound the message. Noise may consist of any unwanted stimulus that renders the message less comprehensible. For example, on the modern battlefield, noise may occur because of conflicting information, irrelevant information, or competing sources of information.

Since Shannon and Weaver's early work, other models of the communication process have been proposed, addressing the weaknesses of the five-step process. Some of these models reflected the increasingly complex nature of team communication. As time went on, network models of communication emerged, further increasing the complexity (and therefore the model validity) of representations of the human communication process. When dealing with distributed decision-making in teams, these models must replace the presumably perfect connections between communicating agents. However, as with shared situational awareness, the research on this topic is just in its beginnings.

8.6 Case Study

In this study, a multi-resolution coordinated mission for Unmanned Air Vehicles (UAV, which are airplanes that are flying without a human pilot on board) is being considered. The C4I system is represented with yet another simulation developed in Matlab/Simulink environment. The model is called MUAV, which is a collaborative UAV testbed (Niland 2006). The agent-augmented multisimulation based decision-making scenario examined in this scenario involves an operator that interacts with both the MUAV software that represents the C4I system and the multisimulation. Figure 8.8 presents the major components of the simulation, which is based on the High Level Architecture (HLA) and its common information infrastructure,

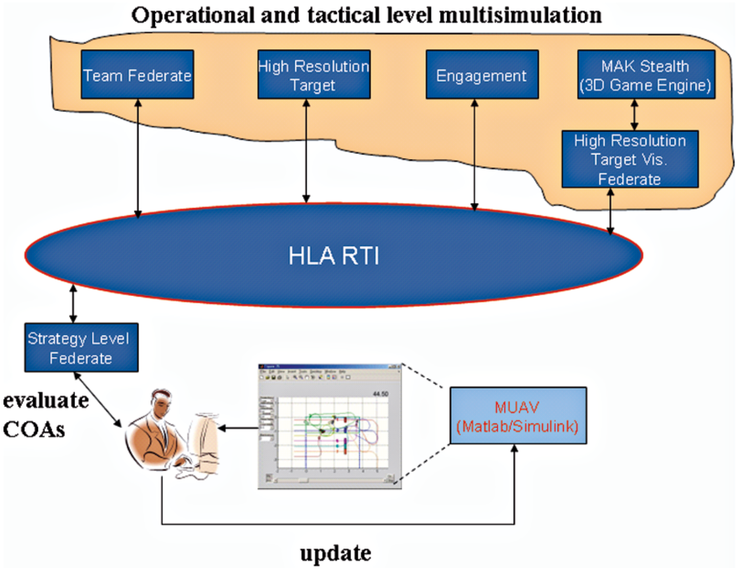


Fig. 8.8. UAV coordination mission study

the Run-Time Infrastructure (RTI). HLA is an international standard for distributed simulation (IEEE 1516–2000).

The scenario starts at the low resolution with a number UAVs sweeping an area that contains multiple targets. Targets are classified as low resolution (i.e., tank battalions) and high-resolution entities (i.e., individual tanks). Individual UAVs can detect and destroy high-resolution entities such as tanks. However, in the case of a detection of an aggregate entity such as a battalion, UAVs aggregate into teams by virtue of a team formation strategy to establish multi-resolution entities, called Teams. The strategy level federate uses inputs of from the operator to (1) cluster entities to identify aggregates and (2) uses agent based team formation protocol, called contract-net, to establish teams. Next, applicable strategies or COAs are recommended by the operator so that teams at the operational and tactical simulation level can be configured by the appropriate behavioral model. If more than one COA is applicable then multiple simulations are initiated, as shown in Fig. 8.9. The multiple simulations at the operational level include behavior from High-resolution Team (HRT), the engagement that represents the tactical strategy used to engage with the targets at the high resolution simulation, the targets, and the visualization behavior. For the low-resolution on higher tactical level, a Matlab/Simulink simulation was used. For the high-resolution simulation of HRT, the MAK Stealth (3D Game Engine) off-the-shelf software was used.

The tactical federate uses intelligent agent support to configure the HRT of a given operational simulation with any of the following strategies. As such, the Coordination Strategy lets the COA protocol vary independently

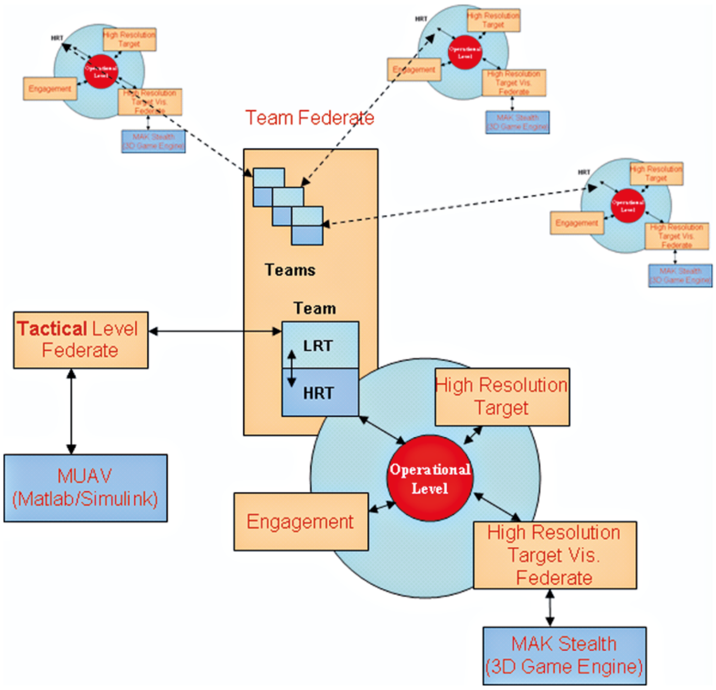


Fig. 8.9. Multiple simulations at the operational level

of the team that uses it. The possibility of configuring teams with multiple COAs enables performing multisimulation, where each simulation facilitates exploring the efficiency and effectiveness of a specific COA. For instance, in our case study we considered two COAs for sweeping the battlefield: Region and Fringe strategy. Figure 8.10 presents the rules of the region strategy, whereas Fig. 8.11 illustrates the rules of Fringe Point Strategy.

In our study, staging from one strategy to another based on the observed conditions is as critical as initiating multiple simulations in the first place. Fig. 8.9 presents demonstrates the connections between HRT and Strategy Federate via a Low Resolution Team (LRT) that coexists with HRT encapsulated within a Multi-resolution Team entity. LRT uses observer agents to monitor the HRT to evaluate the state of the engagement. Corresponding to the time path of the change of a problem should be a time path of the appropriate submodel families. But, the question is what should be the sequence of this shift pattern of models of family? Or should there be trigger mechanisms indicating when a shift should occur? The tactical federate uses an anticipator agent defined in terms of a Bayesian model to decide the correct strategy and instructs the Multi-resolution team to reconfigure its HRT with the selected strategy.

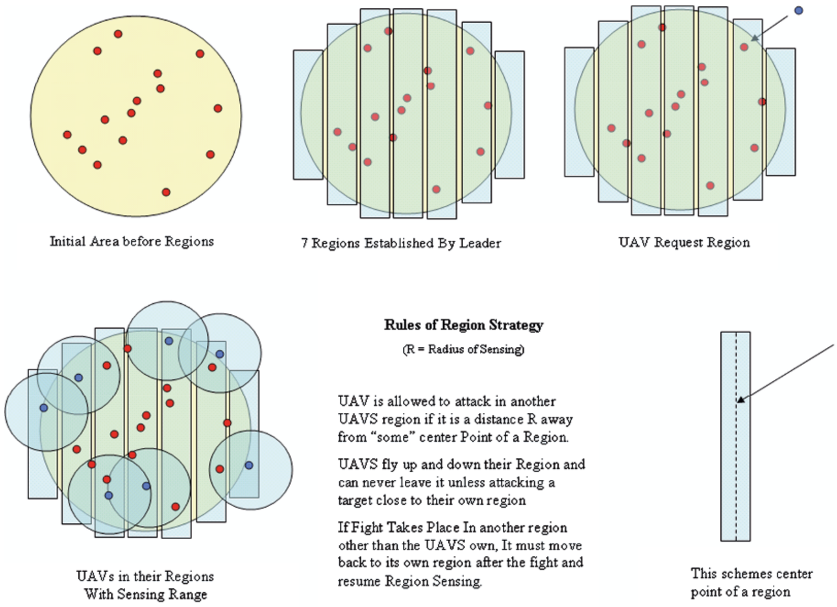


Fig. 8.10. Presentation of the rules of the region strategy

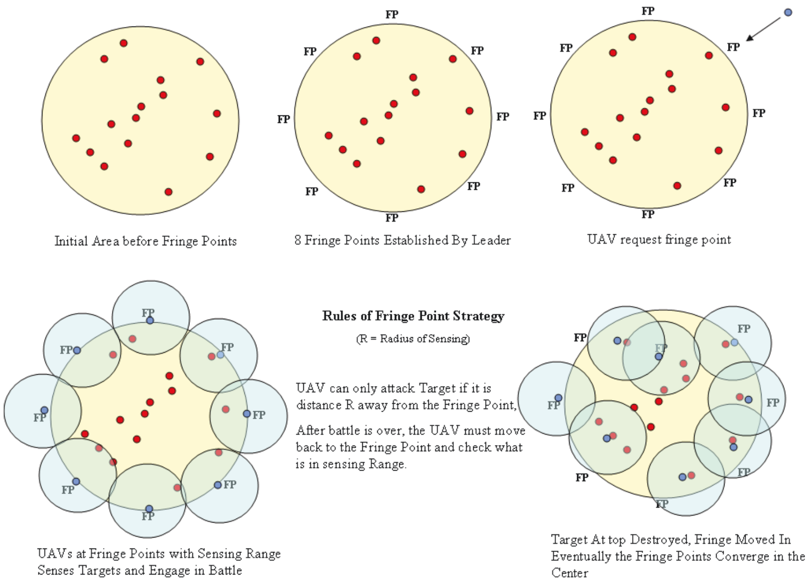


Fig. 8.11. Illustration of the fringe point strategy

8.7 Conclusions

The use of intelligent agents in decision support is common. However, existing work on agent-based decision support mostly focuses on rational choice models, where agents are programmed to seek optimal utilities during negotiation and bargaining. Recent advancements in decision science suggest that pursuing synthesis of alternative decision styles within a coherent framework could have profound effects on the approach to decision-support. Empirical studies of Eisenhardt and Zbaracki (1992) involving mid-to high-level strategic decision makers found that context and environmental circumstances effect the decision-making style employed by the decision makers. In most decision-making scenarios, the nature of the problem changes as the problem unfolds. Initial parameters, as well as scenarios can be irrelevant (i.e., real-time training scenarios) under emergent conditions. Relevant contingency models need to be identified and instantiated to continue exploration. Another aspect that is currently under research, in particular in the ontological community and composability researcher, is the question how model families and multimodels that comprise multi-resolution models (which are models that vary in scope, structure, or resolution) can be used in an orchestrated way in support of decision support. First results are summarized in (Tolk et al. 2007, Tolk et al. 2008), but the research and discussion is ongoing.

In this paper, we suggested a multi-model framework that that subsumes multiple submodels that together constitute the behavior of a complex multi-phased decision-making process. Three distinct decision styles are embedded within a horizontal agent-based decision-support system architecture. Strategies and design considerations for developing experience-based, practical reasoning, and deductive rational choice models of decision-making are examined. It has been argued that situation awareness is a critical component of Naturalistic Decision-making style that is based on experience based reasoning. Perception, understanding, and anticipation mechanisms are discussed as three major subsystems in realizing situation awareness model. These methods and technology will contribute to make agent-based simulation a valuable tool for decision support systems, as their support will become more flexible, credible, and configurable to the users needs.

References

Banks JS and Sundaram RK (1990) Repeated games, finite automata, and complexity. Games and Economic Behavior, 2, pp 97–117

Bigelow JH and Davis PK (2003) Implications for model validation of multi-resolution, multi-perspective modeling (MRMPM) and exploratory Analysis. California, CA: RAND Publication

BISC-SIG (2004) Special Interest Group in Anticipatory Systems. BISC (Berkeley Initiative in Soft Computing) Available online via <http://www.anticipation.info/> (accessed on April 9, 2004)

Bloom A (1968) The republic of Plato (translated with notes and an interpretive essay), New York, NY: Basic Books

Cannon-Bowers J, Salas E and Converse S (1993) Shared mental models in expert team decision making. In Castellan NJ (Ed) Individual and group decision making: Current issues, Hillsdale, NJ: Erlbaum, pp 221–246

Daly JJ and Tolk A (2003) Modeling and simulation integration with network-centric command and control architectures, Fall Simulation Interoperability Workshop, IEEE CS Press

Davis PK, Kulick J and Egner M (2005) Implications of modern decision science for military decision-support systems. California, CA: RAND Publications

Eisenhardt KM and Zbaracki JM (1992) Strategic decision making. Strategic management journal, 13, pp 17–37

Endsley MR (1995) Measure of situation awareness in dynamic systems. Journal for Human Factors, 37(1), pp 65–84

Evan JR and Olson DL (2002) Introduction to Simulation and Risk Analysis (2nd Edition), Prentice Hall, pp 161–179

Ferber J (1999) Multi-agent systems: An introduction to distributed artificial intelligence. New York, NY: Addison-Wesley

Fraser MN and Hipel WK (1984) Conflict analysis models and resolutions. Amsterdam: Elsevier Science Publishers

Gaba DM and Howard SK (1995) Situation awareness in anesthesiology. Journal for Human Factors, 37, pp 20–31

Geyer F and van der Zouven J (1998). Bibliography on sociocybernetics (3rd edition). (http://www.unizar.es/sociocybernetics/bibliografia.html, 1998)

Hamm RM (1988) Moment by moment variation in expert's analytic and intuitive cognitive activity. IEEE Transactions on Systems, Man, and Cybernetics,18(5), pp 757–776

Hammond KR (1986) A theoretically-based review of theory and research in judgement and decision making. Technical report CRJP 260, Center for Research on Judgement and Policy, University of Colorado

IEEE 1516–2000 Family of Standards for Modeling and Simulation (M&S) High Level Architecture (HLA) – (a) IEEE 1516–2000 Framework and Rules; (b) IEEE 1516.1–2000 Federate Interface Specification; (c) IEEE 1516.2–2000 Object Model Template (OMT) Specification

Klein G (1997) The recognition-primed decision (RPD) model: Looking back, looking forward. In Zsambok EC and Klein G (eds) Naturalistic decision making, Mahwah, NJ, Lawrence Erlbaum Associates Publishers, pp 285–292

Knight DE, Curtis WH and Fogel LJ (eds) (1991) Cybernetics, simulation, and conflict resolution, Proc. of the 3rd Annual Symposium of the American Society for Cybernetics (Spartan Books, New York)

Leimar O (1997) Repeated games: A state space approach. Journal of Theoretical Biology, 184, pp 471–498

Murch R and Johnson T (1998) Intelligent Software Agents. Englewood Cliffs, NJ: Prentice Hall

von Neumeann J and Morgenstern O (1953) Theory of games and economic behavior. (3rd ed) Princeton, NJ, Princeton University Press

Niland MW (2006) The migration of a collaborative UAV testbed into the FLAMES simulation environment. In Proceedings of the 2006 Winter Simulation Conference, pp 1266–1272

Ören TI (1987) Model update: A model specification formalism with a generalized view of discontinuity. In Proceedings of the summer computer simulation conference, Montreal, Quebec, Canada, 1987 July 27–30, 689–694

Ören TI (1991) Dynamic templates and semantic rules for simulation advisors and certifiers. In Fishwick AP and Modjeski BR (eds) Knowledge-based simulation: Methodology and application, Berlin, Heidelberg, New York: Springer, pp 53–76

Ören TI (2001) Towards a modelling formalism for conflict management. In Sarjoughian SH and Cellier EF (eds) Discrete-event modeling and simulation: A tapestry of systems and AI-based theories and methodologies, Berlin, Heidelberg, New York: Springer, pp 93–106

Parsons S and Wooldridge M (2002) An introduction to game theory and decision theory. In Parson et al. (eds) Game theory and decision theory in agent-based systems, Kluwer Academic Publishers, pp 1–28

Power D (2002) Decision support systems: Concepts and resources for managers; Chapter 10: Building Model-Driven Decision Support Systems, Quorum Books division Greenwood Publishing, pp 172ff

Rosen R (1985) Anticipatory systems – Philosophical, mathematical and methodological foundations, New York, NY: Pergamon

Sallach D (2003) Interpretive agents. In Proceedings of the Agent 2003: Challenges in Social Simulation Conference, University of Chicago

Shannon, CE and Weaver, W (1949) The mathematical theory of communication. Urbana: University of Illinois Press

Shubik M (ed) (1964) Game theory and related approaches to social behavior. New York, NY, Wiley

Simon H (1982) Models of bounded rationality, vol 1, Cambridge, MA, MIT

Sokolowski JA (2003) Enhance Decision Modeling Using Multiagent System Simulation, SIMULATION, vol 79 no 4, pp. 232–242

Tolk A (2004) An agent-based decision support system architecture for the military domain. In: Phillips-Wren GE and Jain LC (eds) Intelligent decision support systems in agent-mediated environments, Volume 115 Frontiers in Artificial Intelligence and Applications, IOS Press, pp 187–205

Tolk A and Kunde D (2003) Decision support systems in the military environment., In Tonfoni G. and Jain L. (Eds) Innovations in decision support systems, International Series on Advanced Intelligence, Advanced Knowledge International, Magill, Adelaide, Australia, pp 175–210

Tolk A and Gaskins RC (2006) Challenges and Potential of Service-oriented Architectures for Net-Centric Operations. NATO Report MP-HFM-136, Research and Technology Organization: Human Factors and Medicine Panel, Brussels

Tolk A, Diallo SY and Turnitsa CD (2007) Model-Based Data Engineering: Preparing a Paradigm Shift towards Self-Organizing Information Exchange. Proceedings of the Summer Computer Simulation Conference (SCSC-07), pp 1112–1119

Tolk A, Turnitsa CD and Diallo SY, "Implied Ontological Representation within the Levels of Conceptual Interoperability Model," International Journal for Intelligent Decision Technologies 2 (1), (in press)

Tversky A and Kahneman D (1974) Judgment under uncertainty: Heuristics and biases. Science 185, pp 1124–1131

Wikipedia: (Phi-Per) (2004) Philosophy of Perception. Available online via <http://en.wikipedia.org/wiki/Philosophy_of_perception>

Wooldridge M (2002) An introduction to multi-agent systems, New York, NY, Wiley

Yilmaz L and Ören T (2004) Dynamic model updating in simulation with multi-models: A taxonomy and generic agent-based architecture. In Proceedings of the Summer Computer Simulation Conference (SCSC-04), pp 3–8

Yilmaz L, Ören TI and Ghasem-Aghaee N (2006) Simulation-based problem solving environments for conflict studies: Toward exploratory multisimulation with dynamic simulation updating. Simulation, and Gaming Journal 37(4), 534–556

Zachary W (1998) Decision-support systems: Designing to extend the cognitive limits. In Helander GM (ed) Handbook of Human-computer interaction, Amsterdam: Elsevier Science Publishers

Zeigler BP (1986) Toward a simulation methodology for variable structure modeling, In Elzas MS, Ören TI and Zeigler BP (eds) Modeling and simulation methodology in the artificial intelligence era, Amsterdam: North Holland

Zeigler PB, Praehofer HP and Kim GT (2000). Theory of modeling and simulation: Integrating discrete event and continuous complex dynamic systems, New York, NY: Academic

Zsambok EC (1997) Naturalistic decision making: Where are we now? In Zsambok EC and Klein G (eds) Naturalistic decision making, Mahwah NJ, Lawrence Erlbaum Associates Publishers, pp 3–16

Part III

Applications: Intelligent Decision Support

9

A Consensus Support System for Group Decision Making Problems with Heterogeneous Information

F. Mata[1], L. Martínez[2], and E. Herrera-Viedma[3]

[1] Department of Computer Science, University of Jaén, Spain fmata@ujaen.es
[2] Department of Computer Science, University of Jaén, Spain martin@ujaen.es
[3] Department of Computer Science and A.I., University of Granada, Spain viedma@decsai.ugr.es

Summary. A group decision making (GDM) problem is a decision process where several decision makers (experts, judges, etc.) participate and try to reach a common solution. In the literature these problems have been solved carrying out a selection process that returns the solution set of alternatives from the preferences given by the experts. In order to achieve an agreement on the solution set of alternatives among the experts, it would be adequate to carry out a consensus process before the selection process. In the consensus process the experts discuss and change their preferences in order to achieve a big agreement. Due to the fact that the experts may belong to different research areas, they may express their preferences in different information domains. In this contribution we focus on the consensus process in GDM problems defined in heterogeneous contexts where the experts express their preferences by means of numerical, linguistic and interval-valued assessments. We propose a consensus support system model to automate the consensus reaching process, which provides two main advantages: (1) firstly, its ability to cope with GDM problems with heterogeneous information by means of the Fuzzy Sets Theory, and, (2) secondly, it assumes the moderator's tasks, figure traditionally presents in the consensus reaching process.

9.1 Introduction

Group decision-making (GDM) problems may be defined as decision situations where two or more experts try to achieve a common solution about a problem taking into account their opinions or preferences.

In the literature we can find many proposals to solve decision problems where experts use the same information domain to express their preferences (Bui 1987; Herrera and Herrera-Viedma 2000; Kacprzyk 1986; Kim et al. 1999). However, in several occasions it may be more suitable that experts express their opinions in different expression domains according to their own

F. Mata et al.: *A Consensus Support System for Group Decision Making Problems with Heterogeneous Information*, Studies in Computational Intelligence (SCI) **97**, 229–257 (2008)
www.springerlink.com

knowledge or nature of the alternatives. For example, if experts belong to different departments (marketing, accounting, psychology, etc.), they may prefer to provide their opinions by using an information domain closer their research topics. Moreover, in a decision problem we can deal with alternatives whose nature is quantitative and others whose nature is qualitative. The first ones can be assessed by means of precise values like crisp values (Kacprzyk 1986; Yager 1988). However, when alternatives are related to qualitative aspects, it may be difficult to qualify them using precise values. In such cases, where the uncertainty is present, the experts can use interval-valued (Kundu 1997; Le Téno and Mareschal 1998) or linguistic values (Herrera and Herrera-Viedma 2000; Yager 1995) to express their preferences. In such situations, the decision problem is defined into a heterogenous context.

Usually GDM problems have been solved carrying out *Selection Processes* where experts obtain the best solution set of alternatives from the preferences given by themselves (Fodor and Roubens 1994; Roubens 1997). However it may happen that some experts consider that their preferences have not been taken into account to obtain the solution, and therefore they do not agree with it. To avoid this situation, it is suitable to carry out a consensus process (see Fig. 9.1) where the experts discuss and change their preferences in order to reach a sufficient agreement before making a decision (Carlsson et al. 1992; Herrera et al. 1996; Herrera-Viedma et al. 2002; Kacprzyk et al. 1997).

Different methods have been proposed to deal with *Selection Processes* in heterogeneous GDM problems in the literature (Delgado et al. 1998; Herrera et al. 2005; Zhang et al. 2004), but there are not defined specific consensus processes for this kind of problems. Consequently, in this chapter we focus on the *Consensus Process* on GDM problems dealing with heterogeneous information.

The consensus is an important area of research in GDM (Bordogna et al. 1997; Bryson 1996; Carlsson et al. 1992; Fan and Chen 2005; Herrera-Viedma et al. 2002; Kacprzyk et al. 1997; Szmidt and Kacprzyk 2003; Yager 1997).

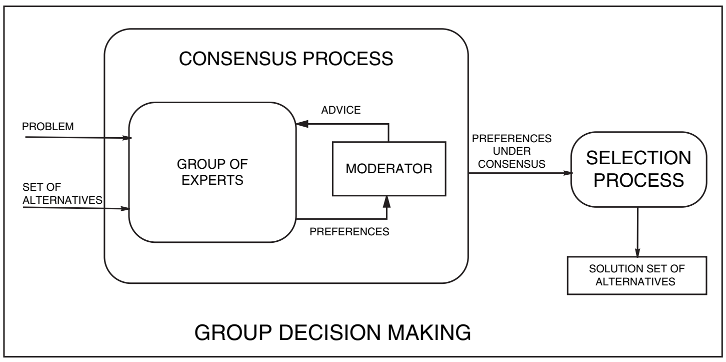


Fig. 9.1. Resolution process of a GDM problem

The consensus is defined as a state of mutual agreement among members of a group where all opinions have been heard and addressed to the satisfaction of the group (Saint and Lawson 1994). The consensus reaching process is a dynamic and iterative process composed of several rounds, where the experts express and discuss their opinions. Traditionally this process is coordinated by a human moderator, who computes the agreement among experts in each round using different consensus measures (Herrera-Viedma et al. 2004; Kuncheva 1991). If the agreement is not enough then the moderator recommends the experts to change their furthest preferences from the group opinion in an effort to make them closer in the next consensus round (Bryson 1996; Zadrozny 1997).

The moderator has been a controversial figure because experts may have complaints about his lack of objectivity. Moreover, in heterogeneous contexts, he may have problems to understand all the different domains and scales in a proper way. Therefore, the aim of this chapter is to present a consensus support system (CSS) model for GDM problems such that:

- The experts can express their preferences by means of linguistic, numerical or interval-valued preference relations, i.e., into a heterogeneous context.
- The moderator's tasks are assumed by an automatic guided advice generator.

The chapter is set out as follows. First, we introduce the GDM problems defined in heterogeneous contexts in the Sect. 2. In the Sect. 3 the different phases of the consensus model are explained in detail. Finally, in the Sect. 4, a practical example is proposed in order to show the performance of the CSS.

9.2 Preliminaries

Let's begin this section introducing the GDM problems based on fuzzy preference relations. Afterwards, it is briefly reviewed different approaches proposed in the literature to express the experts' preferences and finally it is presented the heterogeneous GDM problems.

9.2.1 Group Decision Making Problems

GDM problems are decision situations in which two or more individuals or experts, $E = \{e_1, e_2, \ldots, e_m\}$ $(m \geq 2)$, provide their preferences on a set of alternatives, $X = \{x_1, x_2, \ldots, x_n\}$ $(n \geq 2)$, to derive a solution (an alternative or set of alternatives). Depending on the nature or the knowledge on the alternatives, experts may express their preferences using different approaches.

In fuzzy contexts, experts' preferences are usually expressed by means of fuzzy preference relations (Kacprzyk 1986). A preference relation may be defined as a matrix $P_{e_i} \subset X \times X$

$$\mathbf{P_{e_i}} = \begin{pmatrix} p_i^{11} & \cdots & p_i^{1n} \\ \vdots & \ddots & \vdots \\ p_i^{n1} & \cdots & p_i^{nn} \end{pmatrix},$$

where the value $\mu_{P_{e_i}}(x_l, x_k) = p_i^{lk}$ is interpreted as the preference degree of the alternative x_l over x_k expressed by the expert e_i.

Let's suppose $p^{lj} \in [0,1]$, then:

1. $p^{lj} = 1$ indicates the maximum degree of preference of x_l over x_j.
2. $0.5 \leq p^{lj} \leq 1$ indicates a definitive preference of x_l over x_j.
3. $p^{lj} = 0.5$ indicates the indifference between x_l and x_j.

The fuzzy preference relations may satisfy some of the following properties:

- Reciprocity: $p^{lj} + p^{jl} = 1, \forall l, j$
- Completeness: $p^{lj} + p^{jl} \geq 1, \forall l, j$
- Max–Min Transitivity: $p^{lk} \geq Min(p^{lj}, p^{jk}), \forall l, j, k$
- Max–Max Transitivity: $p^{lk} \geq Max(p^{lj}, p^{jk}), \forall l, j, k$
- Restricted Max–Min Transitivity: $p^{lj} \geq 0.5, p^{lk} \geq 0.5 \Rightarrow p^{lk} \geq Min(p^{lj}, p^{jk})$
- Restricted Max–Max Transitivity: $p^{lj} \geq 0.5, p^{lk} \geq 0.5 \Rightarrow p^{lk} \geq Max(p^{lj}, p^{jk})$
- Additive Transitivity: $p^{lj} + p^{jk} - 0.5 = p^{lk}, \forall l, j, k$

9.2.2 Preferences Modeling

Fuzzy Preference Relations

A valued fuzzy preference relation R on X is defined as a fuzzy subset of the direct product $X \times X$, i.e, $R : X \times X \to [0,1]$. The value, $R(x_l, x_k) = p^{lk}$ denotes the degree in which an alternative x_l is preferred to alternative x_k.

$$\mathbf{P_{e_i}} = \begin{pmatrix} 0.5 & \cdots & 0.7 \\ \vdots & \ddots & \vdots \\ 0.3 & \cdots & 0.5 \end{pmatrix}$$

These were the first type of fuzzy preference relations used in decision making (Kacprzyk 1986) to deal with uncertainty, but soon appeared other approaches to express the uncertainty that will be reviewed in the following subsections.

Interval-Valued Preference Relations

A first approach to add some flexibility to the uncertainty representation problem was by means of interval-valued preferences relations:

$$R : X \times X \to I([0,1]),$$

where $R(x_l, x_k) = p^{lk}$ denotes the interval-valued preference degree of the alternative x_l over x_k. In these approaches (Kundu 1997; Le Téno and Mareschal 1998), the preferences provided by the experts are interval values assessed in $I([0,1])$, where the preference is expressed as $[\underline{a}, \overline{a}]^{lk}$, with $\underline{a} \leq \overline{a}$

$$\mathbf{P_{e_i}} = \begin{pmatrix} [0.5, 0.5] & \cdots & [0.7, 0.9] \\ \vdots & \ddots & \vdots \\ [0.1, 0.3] & \cdots & [0.5, 0.5] \end{pmatrix}.$$

Fuzzy Linguistic Preference Relations

A fuzzy linguistic preference relation is defined as

$$R : X \times X \rightarrow S$$

being $S = (s_0, \ldots, s_g)$ a set of labels.

There are situations in which a better approach to qualify aspects of many activities may be to use linguistic assessments instead of numerical values. The fuzzy linguistic approach represents the information as linguistic values by means of linguistic variables (Zadeh 1975). This approach is adequate to qualify phenomena related to human perception that we often assess using words in natural language. This may arise for different reasons. There are some situations where the information may be unquantifiable due to its nature, and thus, it may be stated only in linguistic terms (e.g., when evaluating the "comfort" or "design" of a car, terms like "bad", "poor", "tolerable", "average", "good" can be used (Levrat et al. 1997)). In other cases, according to (Zadeh 1996) there is a tolerance for imprecision which can be exploited to achieve tractability, robustness, low solution cost, and better rapport with reality (e.g., when evaluating the speed of a car, linguistic terms like "fast", "very fast", "slow" are used instead of numerical values).

We have to choose the appropriate linguistic descriptors for the term set and their semantics. One possibility of generating the linguistic term set consists in directly supplying the term set by considering all terms distributed on a scale on which a total order is defined (Yager 1995). For example, a set of seven terms S, could be given as follows:

$$S = \{s_0 = none, s_1 = very\ low, s_2 = low, s_3 = medium,$$
$$s_4 = high, s_5 = very\ high, s_6 = perfect\}.$$

In these cases, it is usually required that there exist:

1. A negation operator $\text{Neg}(s_i) = s_j$ such that $j = g$-i (g+1 is the cardinality of the term set)
2. A maximization operator: $\text{Max}(s_i, s_j) = s_i$ if $s_i \geq s_j$
3. A minimization operator: $\text{Min}(s_i, s_j) = s_i$ if $s_i \leq s_j$

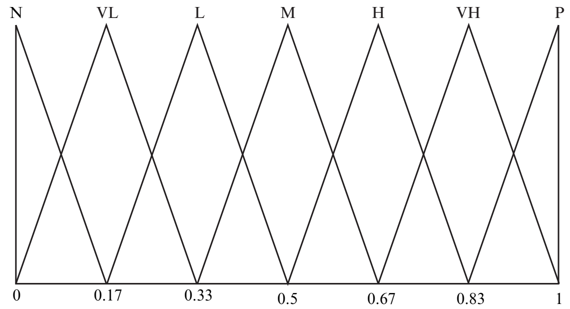


Fig. 9.2. A set of seven linguistic terms with its semantics

The semantics of the terms is given by fuzzy numbers defined in the [0,1] interval. A way to characterize a fuzzy number is to use a representation based on parameters of its membership function (Bonissone and Decker 1986). For example, we may assign the following semantics to the set of seven terms via triangular fuzzy numbers:

$$\begin{array}{ll} Perfect(P) = (0.83, 1, 1) & Very\_High(VH) = (0.67, 0.83, 1) \\ High(H) = (0.5, 0.67, 0.83) & Medium(M) = (0.33, 0.5, 0.67) \\ Low(L) = (0.17, 0.33, 0.5) & Very\_Low(VL) = (0, 0.17, 0.33) \\ None(N) = (0, 0, 0.17), & \end{array}$$

which is graphically shown in Fig. 9.2.

Therefore a linguistic preference relation $R(x_l, x_k)$ denotes the linguistic preference degree of the alternative x_l over x_k. Using the linguistic term set shown in the Fig. 9.2, a linguistic preference relation could be:

$$\mathbf{P_{e_i}} = \begin{pmatrix} M & \cdots & VH \\ \vdots & \ddots & \vdots \\ VL & \cdots & M \end{pmatrix}.$$

9.2.3 Group Decision Making Problems Defined on Heterogeneous Contexts

The ideal situation in a GDM problem is that all experts have a wide knowledge about the alternatives and provide their opinions in a numerical precise scale. However, in some cases, experts may belong to distinct research areas and have different levels of knowledge about the alternatives. Due to this, the experts may prefer to express their preferences by means of different information domains. In such cases, the problem is defined in a heterogeneous context.

In this chapter we deal with heterogenous GDM problems where the experts express their preferences using different expression domains (numerical, interval-valued or linguistic), $D_i \in \{N|I|L\}$. Each expert gives their opinions by means of a fuzzy preference relation defined on an unique expression domain, $\mathbf{P_{e_i}}$:

$$\mathbf{P_{e_i}} = \begin{pmatrix} p_i^{11} & \cdots & p_i^{1n} \\ \vdots & \ddots & \vdots \\ p_i^{n1} & \cdots & p_i^{nn} \end{pmatrix},$$

where $p_i^{lk} \in D_i$ represents the preference of the alternative x_l over the alternative x_k given by the expert e_i.

9.3 A Consensus Support System Model for GDM Problems with Heterogeneous Information

In this section we present the model of a consensus support system for GDM problems with heterogeneous information. The CSS model has two main features:

1. It is able to carry out the consensus process in heterogeneous GDM problems with numerical, interval-valued and linguistic assessments.
2. It includes a guided advice generator that assumes the moderator's tasks and recommends the changes in experts' preferences in order to obtain a high consensus degree.

The CSS model will be built up using:

- A methodology to unify the heterogeneous information into a single domain.
- Different measurements to cope with the agreement: *consensus degree* and *proximity measure*. The first one is used to evaluate the agreement amongst the experts, while the second one is used to measure the distance between the collective and individual expert's preferences.
- A set of advice rules based on the these measurements are used to guide the direction of the changes in the experts' opinions.

The CSS model consists of the following phases depicted in Fig. 9.3:

1. *Making the information uniform.* In this phase, the experts' heterogeneous preferences are unified into an unique domain.
2. *Computing consensus degree.* In this phase consensus degree amongst the experts is computed. To do so, a similarity function is used to calculate the coincidence amongst experts' preferences.
3. *Checking the agreement.* In this phase the CSS controls the level of agreement achieved amongst experts. If the agreement is greater than a specified consensus threshold (γ) then the consensus process will stop and the selection process will be applied to obtain the solution of the problem. Otherwise, in the following phase the experts' preferences must be modified.

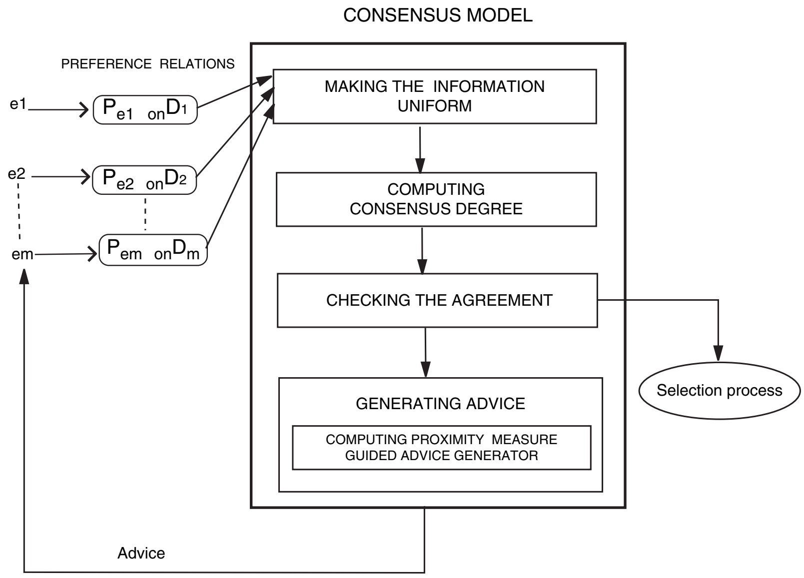


Fig. 9.3. A CSS model with heterogeneous information

4. *Generating advice.* To help experts change their preferences, the CSS generates a set of recommendations or advice. To do this, a proximity measure is used in conjunction with the consensus degree to build a guided advice generator in charge of identifying the preferences to be changed and recommending experts, how should be the changes in order to increase the agreement in the next consensus round.

9.3.1 Making the Information Uniform

Considering that we are dealing with heterogeneous contexts with numerical, interval-valued and linguistic information and because of there are not standard operators to manage directly heterogeneous information, we need to unify this into a common utility space that we will call basic linguistic term set (BLTS), $S_T = \{s_0, \ldots, s_g\}$ (Fig. 9.4). To do so, as it was proposed in (Herrera et al. 2005), we define different transformation functions to transform each numerical, interval-valued and linguistic preference value into a fuzzy set defined in BLTS, $F(S_T)$.

Transforming Numerical Values in $[0,1]$ into $F(S_T)$

To transform a numerical value into a fuzzy set on S_T, we use the following function. Let ϑ be a numerical value, $\vartheta \in [0,1]$, and $S_T = \{s_0, \ldots, s_g\}$ the BLTS. The function τ_{NS_T} that transforms a numerical value ϑ into a fuzzy set on S_T is defined as (Herrera and Martínez 2000):

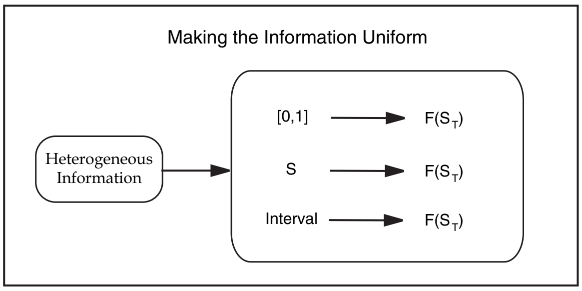


Fig. 9.4. Unification process of heterogeneous information

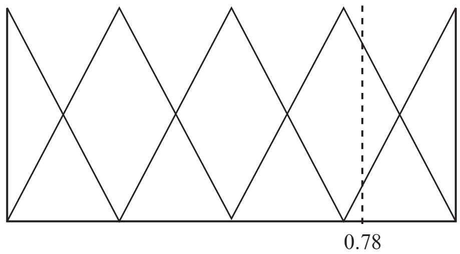


Fig. 9.5. Transforming a numerical value into a fuzzy set in S

$$\tau_{NS_T} : [0,1] \rightarrow F(S_T)$$
$$\tau_{NS_T}(\vartheta) = \{(s_0, \gamma_0), \ldots, (s_g, \gamma_g)\}, s_i \in S_T \, and \, \gamma_i \in [0,1]$$
$$\gamma_i = \mu_{s_i}(\vartheta) = \begin{cases} 0, & if\, \vartheta \notin Support(\mu_{s_i}(x)) \\ \frac{\vartheta - a_i}{b_i - a_i}, & if\, a_i \leq \vartheta \leq b_i \\ 1, & if\, b_i \leq \vartheta \leq d_i \\ \frac{c_i - \vartheta}{c_i - d_i}, & if\, d_i \leq \vartheta \leq c_i \end{cases}$$

Remark 1. We consider membership functions, $\mu_{s_i}(\cdot)$, for linguistic labels, $s_i \in S_T$, represented by a parametric function (a_i, b_i, d_i, c_i). A particular case are the linguistic assessments whose membership functions a triangular, i.e., $b_i = d_i$.

Example 1 *Let $\vartheta = 0.78$ be a numerical value to be transformed into a fuzzy set in $S = \{s_0, \ldots, s_4\}$. The semantic of this term set is:*

$$s_0 = (0, 0, 0.25), s_1 = (0, 0.25, 0.5), s_2 = (0.25, 0.5, 0.75), s_3 = (0.5, 0.75, 1)$$
$$s_4 = (0.75, 1, 1)$$

Therefore, the fuzzy set obtained is (see Fig. 9.5):

$$\tau_{NS_T}(0.78) = \{(s_0, 0), (s_1, 0), (s_2, 0), (s_3, 0.88), (s_4, 0.12)\}$$

Transforming Linguistic Terms in S into $F(S_T)$

To transform a linguistic value into a fuzzy set on S_T, we use the following function. Let $S = \{l_0, \ldots, l_p\}$ and $S_T = \{s_0, \ldots, s_g\}$ be two linguistic term sets, such that, $g \geq p$. Then, the function τ_{SS_T} that transforms $l_i \in S$ into a fuzzy set on S_T is defined as:

$$\begin{gathered} \tau_{SS_T} : S \rightarrow F(S_T) \\ \tau_{SS_T}(l_i) = \{(s_k, \gamma_k^i) \,/\, k \in \{0, \ldots, g\}\}, \; \forall l_i \in S \\ \gamma_k^i = \max_y \min\{\mu_{l_i}(y), \mu_{s_k}(y)\}, \end{gathered}$$

where $F(S_T)$ is the set of fuzzy sets defined in S_T, and $\mu_{l_i}(\cdot)$ and $\mu_{s_k}(\cdot)$ are the membership functions of the fuzzy sets associated with the terms l_i and s_k, respectively.

Therefore, the result of τ_{SS_T} for any linguistic value of S is a fuzzy set defined in S_T.

Remark 2. In the case that the linguistic term set S of the non-homogeneous contexts let be chosen as S_T then the fuzzy set that represents a linguistic term will be all **0** except the value correspondent to the ordinal of the linguistic label that will be **1**.

Example 2 *Let $S = \{l_0, l_1, \ldots, l_4\}$ and $S_T = \{s_0, s_1, \ldots, s_6\}$ be two term set, with 5 and 7 labels, respectively, and with the following semantics associated:*

$$\begin{array}{ll} l_0 = (0, 0, 0.25) & s_0 = (0, 0, 0.16) \\ l_1 = (0, 0.25, 0.5) & s_1 = (0, 0.16, 0.34) \\ l_2 = (0.25, 0.5, 0.75) & s_2 = (0.16, 0.34, 0.5) \\ l_3 = (0.5, 0.75, 1) & s_3 = (0.34, 0.5, 0.66) \\ l_4 = (0.75, 1, 1) & s_4 = (0.5, 0.66, 0.84) \\ & s_5 = (0.66, 0.84, 1) \\ & s_6 = (0.84, 1, 1) \end{array}$$

The fuzzy set obtained after applying τ_{SS_T} for l_1 is (see Fig. 9.6):

$$\tau_{SS_T}(l_1) = \{(s_0, 0.39), (s_1, 0.85), (s_2, 0.85), (s_3, 0.39), (s_4, 0), (s_5, 0), (s_6, 0)\}.$$

Transforming Interval-Valued into $F(S_T)$

To transform an interval-valued into a fuzzy set on S_T, we use the following function. Let $I = [\underline{i}, \overline{i}]$ an interval valued in $[0, 1]$ and $S_T = \{s_0, \ldots, s_g\}$ the BLTS . Then, the function τ_{IS_T} that transforms the interval-valued I into a fuzzy set on S_T is defined as:

$$\begin{gathered} \tau_{IS_T} : I \rightarrow F(S_T) \\ \tau_{IS_T}(I) = \{(s_k, \gamma_k^i) \,/\, k \in \{0, \ldots, g\}\}, \\ \gamma_k^i = \max_y \min\{\mu_I(y), \mu_{s_k}(y)\}, \end{gathered}$$

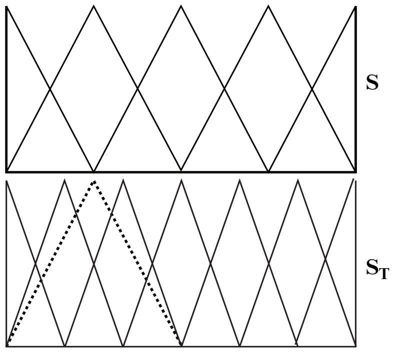


Fig. 9.6. Transforming $l_1 \in S$ into a fuzzy set in S_T

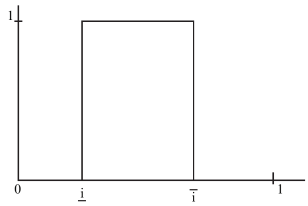


Fig. 9.7. Membership function of $I = [\underline{i}, \overline{i}]$

where $F(S_T)$ is the set of fuzzy sets defined in S_T, and $\mu_I(\cdot)$ and $\mu_{s_k}(\cdot)$ are the membership functions associated with the interval-valued I and terms s_k, respectively.

Remark 3. We assume that the interval-valued has a representation inspired in the membership function of fuzzy sets (Kuchta 2000):

$$\mu_I(\vartheta) = \begin{cases} 0, \, if \, \vartheta < \underline{i} \\ 1, \, if \, \underline{i} \leq \vartheta \leq \overline{i}, \\ 0, \, if \, \overline{i} < \vartheta \end{cases}$$

where ϑ is a value in $[0, 1]$. In Fig. 9.7 can be observed the graphical representation of an interval.

Example 3 *Let $I = [0.6, 0.78]$ be an interval-valued to be transformed into a fuzzy set in S_T with five terms symmetrically distributed. The fuzzy set obtained after applying τ_{IS_T} is (see Fig. 9.8):*

$$\tau_{IS_T}([0.6, 0.78]) = \{(s_0, 0), (s_1, 0), (s_2, 0.6), (s_3, 1), (s_4, 0.2)\}.$$

Results of the Unification Process

Once we have introduced in the previous subsections each one of the different transformation functions, to note that after the unification process and

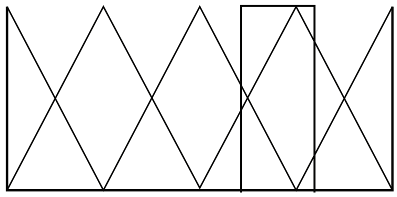


Fig. 9.8. Transforming $[0.6, 0.78]$ into a fuzzy set in S_T

assuming that each fuzzy set will be shown by means of its membership degrees $(\alpha_{i0}^{lk}, \ldots, \alpha_{ig}^{lk})$, the preferences of each expert will be represented as a matrix of fuzzy sets, $\tilde{\mathbf{P}}_{\mathbf{e_i}}$:

$$\tilde{\mathbf{P}}_{\mathbf{e_i}} = \begin{pmatrix} \tilde{p}_i^{11} = (\alpha_{i0}^{11}, \ldots, \alpha_{ig}^{11}) & \cdots & \tilde{p}_i^{1n} = (\alpha_{i0}^{1n}, \ldots, \alpha_{ig}^{1n}) \\ \vdots & \ddots & \vdots \\ \tilde{p}_i^{n1} = (\alpha_{i0}^{n1}, \ldots, \alpha_{ig}^{n1}) & \cdots & \tilde{p}_i^{nn} = (\alpha_{i0}^{nn}, \ldots, \alpha_{ig}^{nn}) \end{pmatrix}.$$

9.3.2 Computing Consensus Degree

The consensus degree evaluates the level of existent agreement among the experts. So, if experts' preferences are similar, the consensus degree will be high, else, if preferences are very different, the consensus degree will be low. To compute the level of agreement, a consensus matrix is obtained aggregating the values which represent the similarity or distance among the experts' preferences, comparing each other.

The distance between two preferences $\tilde{p}_i^{lk}$ and $\tilde{p}_j^{lk}$ is computed by means of the similarity function $s(\tilde{p}_i^{lk}, \tilde{p}_j^{lk})$ measured in the unit interval $[0, 1]$ (Herrera-Viedma et al. 2005):

$$s(\tilde{p}_i^{lk}, \tilde{p}_j^{lk}) = 1 - \left| \frac{cv_i^{lk} - cv_j^{lk}}{g} \right|. \tag{9.1}$$

The cv_i^{lk} is the central value of the fuzzy set:

$$cv_i^{lk} = \frac{\sum_{h=0}^{g} index(s_h^i) \cdot \alpha_{ih}^{lk}}{\sum_{h=0}^{g} \alpha_{ih}^{lk}}, \tag{9.2}$$

and represents the average position or center of gravity of the information contained in the fuzzy set $p_i^{lk} = (\alpha_{i0}^{lk}, \ldots, \alpha_{ig}^{lk})$, being $index(s_h^i) = h$. The range of this central value is the closed interval $[0, g]$.

The closer $s(\tilde{p}_i^{lk}, \tilde{p}_j^{lk})$ to 1 the more similar preferences p_i^{lk} and p_j^{lk} are, while the closer $s(\tilde{p}_i^{lk}, \tilde{p}_j^{lk})$ to 0 the more distant p_i^{lk} and p_j^{lk} are.

Once we have defined the function to evaluate the similarity, the consensus degree is computed according to the following steps:

1. First, the central values of all fuzzy sets are calculated:

$$cv_i^{lk};\ \forall\ i = 1, \ldots, m;\ \ l, k = 1, \ldots, n\ \wedge\ l \neq k. \tag{9.3}$$

2. Afterwards, for each pair of experts e_i and e_j $(i < j)$, a *similarity matrix* $SM_{ij} = \left(sm_{ij}^{lk}\right)$ is calculated, where

$$sm_{ij}^{lk} = s(\tilde{p}_i^{lk}, \tilde{p}_j^{lk}). \tag{9.4}$$

3. Finally a *consensus matrix*, CM, is obtained by aggregating all the similarity matrices

$$CM = \begin{pmatrix} cm^{11} & \cdots & cm^{1n} \\ \vdots & \ddots & \vdots \\ cm^{n1} & \cdots & cm^{nn} \end{pmatrix}.$$

   This aggregation is carried out at the level of pairs of alternatives:

$$cm^{lk} = \phi(sm_{ij}^{lk});\ i, j = 1, \ldots, m\ \ \wedge \forall\ l, k = 1, \ldots, n\ \wedge\ i < j.$$

   In our case, we use the arithmetic mean as the aggregation function ϕ, although, different aggregation operators could be used according to the particular properties we want to implement.

Interpretation of the Consensus Degree

The consensus degree is analyzed in three different levels: pairs of alternatives, alternatives and relations. In this way, we can know in a precise way the level of agreement in each pair and so to identify the pairs as well as the alternatives in which there exists greater disagreement.

Level 1. *Consensus on pairs of alternatives.* The consensus degree on a pair of alternatives (x_l, x_k), called cp^{lk}, measures the consensus degree amongst all the experts on that pair. In our case, this is expressed by the element (l, k) of the consensus matrix CM, i.e.,

$$cp^{lk} = cm^{lk},\ \ \forall l, k = 1, \ldots, n\ \ \wedge\ \ l \neq k.$$

Values of cp^{lk} close to 1 mean a greater agreement. This measure will allow the identification of those pairs of alternatives with a poor level of agreement.

Level 2. *Consensus on alternatives.* The consensus degree on an alternative x_l, called ca^l, measures the consensus degree amongst all the experts on that alternative. It is calculated as the average of each row l of the consensus matrix CM, i.e.,

$$ca^l = \frac{\sum_{k=1,\ l \neq k}^{n} cp^{lk}}{n - 1}. \tag{9.5}$$

These values are used to propose the modification of preferences associated to those alternatives with a consensus degree lower than a minimal consensus threshold γ, i.e., $ca^l < \gamma$.

Level 3. *Consensus on relations or global consensus.* The consensus degree on relations, called cr, measures the global consensus degree amongst the experts' preferences. It is computed as the average of all the consensus degrees on the alternatives, i.e.,

$$cr = \frac{\sum_{l=1}^{n} ca^l}{n}. \tag{9.6}$$

The CSS uses this value to check the level of agreement achieved in each round, so if cr is closer to 1, the level of agreement is high, while if cr is closer to 0, the level of agreement is low.

9.3.3 Checking the Agreement

In this phase the CSS controls the level of agreement achieved in the current consensus round. Before applying the CSS model, a minimum consensus threshold, $\gamma \in [0, 1]$, is fixed, which will depend on the particular problem we are dealing with. When the consequences of the decision are of a transcendent importance, the minimum level of consensus required to make that decision should be logically high, for example $\gamma = 0.8$ or higher. At the other extreme, when the consequences are not so transcendental (but are still important) and it is urgent to obtain a solution of the problem, a fewer consensus threshold near to 0.5 could be required.

In any case, independently of the value γ, when the global consensus cr reaches γ, the CSS will stop and the selection process will be applied to obtain the solution. However, there is the possibility that the global consensus will not converge to consensus threshold and the process will get block. In order to avoid this circumstance, the model incorporates a parameter, $Maxcycles$, to limit the number of consensus rounds to carry out. The performance of this phase is shown in Fig. 9.9.

9.3.4 Generating Advice

When the agreement is not big enough, $cr < \gamma$, experts should modify their preferences in order to make them closer and increase the consensus in the next consensus round. To do so, we will use proximity measures to identify the furthest experts' preferences from the collective opinion. Once these preferences have been identified, a guided advance generator is in charge of suggesting how to change them in order to increase the consensus in the next consensus round. Both processes are presented in detail following.

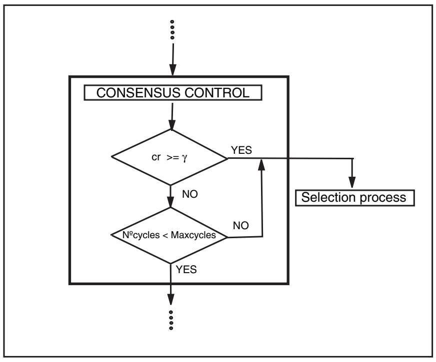


Fig. 9.9. Consensus control

Computing Proximity Measure

The proximity measure evaluates the distance between the individual experts' preferences and the collective preference. To calculate it, firstly we need to obtain a collective preference relations $\tilde{\mathbf{P}}_c$,

$$\tilde{\mathbf{P}}_{\mathbf{c}} = \begin{pmatrix} \tilde{p}_c^{11} & \cdots & \tilde{p}_c^{1n} \\ \vdots & \ddots & \vdots \\ \tilde{p}_c^{n1} & \cdots & \tilde{p}_c^{nn} \end{pmatrix}$$

which represents the group's opinion. $\tilde{\mathbf{P}}_{\mathbf{c}}$ is calculated by aggregating the set of (uniformed) individual preference relations $\{\tilde{\mathbf{P}}_{\mathbf{e_1}}, \ldots, \tilde{\mathbf{P}}_{\mathbf{e_m}}\}$:

$$\tilde{p}_c^{lk} = \psi(\tilde{p}_1^{lk}, \ldots, \tilde{p}_m^{lk}) = (\alpha_{c0}^{lk}, \ldots, \alpha_{cg}^{lk}),$$

where

$$\alpha_{cj}^{lk} = \psi(\alpha_{1j}^{lk}, \ldots, \alpha_{mj}^{lk})$$

being ψ an "aggregation operator".

Once the CSS has obtained the collective preference relation, it computes a proximity matrix, PM_i, for each expert e_i,

$$PM_i = \begin{pmatrix} pm_i^{11} & \cdots & pm_i^{1n} \\ \vdots & \ddots & \vdots \\ pm_i^{n1} & \cdots & pm_i^{nn} \end{pmatrix}.$$

To evaluate the proximity between each expert's individual preferences, $\tilde{\mathbf{P}}_{\mathbf{e_i}}$, and collective preferences, $\tilde{\mathbf{P}}_{\mathbf{c}}$, we use the similarity function defined in expression (9.1),

$$pm_i^{lk} = s(\tilde{p}_i^{lk}, \tilde{p}_c^{lk}).$$

These matrices contain the necessary information to know the position of the preferences of each expert with regards to the group's position.

Interpretation Proximity Measures

From the proximity matrices we can also know the proximity of the preferences of each expert at level of pairs of alternatives, alternatives and relations. In this way it is easy to identify the furthest experts on those assessments where the consensus is not enough:

Level 1. *Proximity on pairs of alternatives.* Given an expert e_i, his/her proximity measure on a pair of alternatives, (x_l, x_k), called pp_i^{lk}, measures the proximity between his/her preference value and the collective's one on that pair. In our case, this value coincides with the element (l, k) of the proximity matrix PM_i, i.e.,

$$pp_i^{lk} = pm_i^{lk}, \quad \forall l, k = 1, \ldots, n \quad \wedge \quad l \neq k.$$

Level 2. *Proximity on alternatives.* Given an expert e_i, his/her proximity measure on an alternative x_l, called pa^l, measures the proximity between his/her preference values on that alternative and the collective's ones. It is computed as the average of the proximities on pairs of alternatives of x_l

$$pa_i^l = \frac{\sum_{k=1, k \neq l}^{n} pp_i^{lk}}{n-1}. \tag{9.7}$$

Level 3. *Proximity on the relation.* Given an expert e_i, his/her proximity measure on the relation, called pr_i, measures the global proximity between his/her preference values on all alternatives and the collective's one. It is computed as the average of all proximity on alternative values, i.e.,

$$pr_i = \frac{\sum_{l=1}^{n} pa_i^l}{n}. \tag{9.8}$$

If pr_i is close to 1 then e_i contributes positively to the consensus, while if pr_i is close to 0 then e_i has a negative contribution to consensus.

Guided Advice Generator

The goal of the guided advice generator is to identify the furthest experts' preferences and suggest how to change them in order to increase the consensus.

To achieve this purpose the guided advice generator uses two types of advice rules: identification rules and direction rules.

Identification Rules (IR)

These rules identify what experts, alternatives and pairs of alternatives should be changed. In this way, the model only focuses on the preferences in disagreement and will not recommend to change those preferences where the agreement is enough. The model uses three rules:

1. *An identification rule of experts.* It identifies those experts that should change some of their preferences values. Previously, we should have decided the number or % of experts (ne) that should modify their preferences. The choice of the value of ne may depend on the type of problem and/or the amount of time available to achieve the consensus. If a quick achievement of consensus is desired, then the value of ne might be high (for example $ne = 75\%$), while if ne is low (for example $ne = 25\%$) then more time will be needed to reach the consensus. Once decided the number of experts, the ne experts with the lowest proximity values must change their preferences. This set of experts is denoted as $EXPCH$. Therefore, the identification rule of experts is the following:
   IR.1. $\forall e_i \in E \cap EXPCH$, then e_i must change his/her preferences, being
   $$EXPCH = \{e_{\sigma(1)}, \ldots, e_{\sigma(ne)}\},$$
   where σ is a permutation over the set of proximities on the relation defined as $pr_{\sigma(j)} \leq pr_{\sigma(i)} \ \forall \ j \leq i$.
2. *An identification rule of alternatives.* It identifies those alternatives where there is not enough consensus and therefore they should be changed. This set of alternatives is denoted as ALT and is composed of those alternatives whose consensus degree ca^l is lower than the consensus threshold γ, i.e.,
   $$ALT = \{x_l \in X | \ ca^l < \gamma\}.$$
   The identification rule of alternatives is the following:
   IR.2. $\forall e_i \in EXPCH$, e_i should change some assessments associated to the pairs that belong to the alternative x_l, such that, $x_l \in ALT$.
3. *An identification rule of pairs of alternatives.* It identifies those particular pairs of alternatives (x_l, x_k) of the alternatives in disagreement $x_l \in ALT$ that should be changed. This set of pairs of alternatives is denoted as $PALT_i$. To do this, we use the proximity measures on pairs of alternatives, being the identification rule the following:
   IR.3. $\forall (x_l \in ALT \ \wedge \ e_i \in EXPCH)$, if $(x_l, x_k) \in PALT_i$ then e_i should change p_i^{lk}, being $PALT_i$ the set of pairs of alternatives (x_l, x_k) whose proximity values at level of pairs, pp_i^{lk}, are fewer that a minimum proximity threshold, β, i.e.,
   $$PALT_i = \{(x_l, x_k) \mid x_l \in ALT \ \wedge \ e_i \in EXPCH \ \wedge \ pp_i^{lk} < \beta\}.$$
   Clearly, the greater β the greater the number of changes needed.

Direction Rules (DR)

Once the model has identified the pairs of alternatives to be changed, $(x_l, x_k) \in PALT_i$, it uses a set of direction rules to suggest how to change the current assessments in order to increase the agreement in the next consensus round. Taking into account that $\tilde{p}_i^{lk}$ is a fuzzy set, the guided advice generator defines two direction parameters: *ml* or main and *sl* or secondary. These parameters represent the value and position of the two highest membership values of the expert's preference $\left(\tilde{p}_i^{lk}(ml_{pos}), \tilde{p}_i^{lk}(ml_{val}), \tilde{p}_i^{lk}(sl_{pos}), \tilde{p}_i^{lk}(sl_{val})\right)$ and the collective preference $\left(\tilde{p}_c^{lk}(ml_{pos}),\ \tilde{p}_c^{lk}(ml_{val}), \tilde{p}_c^{lk}(sl_{pos}), \tilde{p}_c^{lk}(sl_{val})\right)$. The rules compare the position and value of the parameters *ml* and *sl* of the expert's preference and collective preference. According to the result of this comparison, the advice generator suggests increase o decrease the expert's current assessment.

These parameters are used by the following direction rules:

DR.1. If $\tilde{p}_i^{lk}(ml_{pos}) > \tilde{p}_c^{lk}(ml_{pos})$ then the expert e_i should decrease the assessment associated to the pair of alternatives (x_l, x_k).
DR.2. If $\tilde{p}_i^{lk}(ml_{pos}) < \tilde{p}_c^{lk}(ml_{pos})$ then the expert e_i should increase the assessment associated to the pair of alternatives (x_l, x_k).
DR.3. If $\tilde{p}_i^{lk}(ml_{pos}) = \tilde{p}_c^{lk}(ml_{pos})$ then rules DR.1, and DR.2 are applied using the membership values of the main labels, $\tilde{p}_i^{lk}(ml_{val})$ and $\tilde{p}_c^{lk}(ml_{val})$.
DR.4. If $\left(\tilde{p}_i^{lk}(ml_{pos}) = \tilde{p}_c^{lk}(ml_{pos}), \tilde{p}_i^{lk}(ml_{val}) = \tilde{p}_c^{lk}(ml_{val})\right)$, then rules DR.1, DR.2, and DR.3 are applied using the position and membership values of the secondary labels *sl*.

Example 4 *Given the expert's preference,* $\tilde{p}_1^{12} = (\underline{\boldsymbol{1}}, \underline{\boldsymbol{0.67}}, 0.33, 0, 0, 0, 0, 0, 0)$, *and the collective preference* $\tilde{p}_c^{12} = (\underline{\boldsymbol{0.38}}, 0.28, 0.14, \overline{0.17}, \underline{\boldsymbol{0.3}}, 0.27, 0.19, 0.11, 0.13)$, *their direction parameters are respectively:*

$$\tilde{p}_1^{12}(ml_{pos}) = 0, \quad \tilde{p}_1^{12}(ml_{val}) = 1, \qquad \tilde{p}_1^{12}(sl_{pos}) = 1, \quad \tilde{p}_1^{12}(sl_{val}) = 0.67,$$

$$\tilde{p}_c^{12}(ml_{pos}) = 0, \quad \tilde{p}_c^{12}(ml_{val}) = 0.38, \qquad \tilde{p}_c^{12}(sl_{pos}) = 4, \quad \tilde{p}_c^{12}(sl_{val}) = 0.3.$$

Finally to note that the consensus reaching process will depend on the size of the group of experts as well as on the size of the set of alternatives. So, when these sizes are small and when opinions are similar, the consensus level required is easier to obtain. However, when the experts opinions are quite different, the number of consensus rounds and the time to reach the wanted agreement will be greater.

9.4 Example of Application of the CSS Model

In this section we show an application example of the proposed CSS model to carry out a consensus reaching process in a real-word problem. We shall only focus on the consensus process, by recommending readers to consult (Delgado et al. 1998; Herrera and Martínez 2000; Herrera et al. 2005) to see how the selection of the best alternative(s) is carried out.

A drink company specializing in sport drinks is planning to launch a new soft drink, but first, it has to choose a taste that is accepted by the majority of the sportsmen. The company is considering four possible tastes:

- Lemon taste: x_1
- Apple taste: x_2
- Orange taste: x_3
- Peach taste: x_4

The management has decided to consult three experts. Experts have to express their preferences about the different tastes or alternatives by using preferences relations and they must reach a high level of agreement before making the decision. Each expert belongs to a different area and expresses his preferences by using a different information domain:

- The expert e_1 belongs to the marketing department and gives his preferences by means of numerical values in $[0,1]$, $\mathbf{P_{e_1}}$.
- The expert e_2 is an elite sportsman and prefers to use linguistic assessments of the linguistic term set S described in section "Fuzzy Linguistic Preference Relations" $\mathbf{P_{e_2}}$.
- The expert e_3 is a specialistic in soft drinks and gives his preferences by means interval-valued preference values in $[0,1]$, $\mathbf{P_{e_3}}$.

Note that the preferences p_i^{ll} do not have been considered because they represent the preference degree of an alternative over itself

$$\mathbf{P_{e_1}} = \begin{pmatrix} - & .5 & .8 & .4 \\ .3 & - & .9 & .3 \\ .3 & .2 & - & .4 \\ .9 & .8 & .5 & - \end{pmatrix}; \ \mathbf{P_{e_2}} = \begin{pmatrix} - & H & VH & M \\ L & - & H & VH \\ VL & N & - & VH \\ L & VL & N & - \end{pmatrix}$$

$$\mathbf{P_{e_3}} = \begin{pmatrix} - & [.7,.8] & [.65,.7] & [.8,.9] \\ [.3,.35] & - & [.6,.7] & [.8,.85] \\ [.3,.35] & [.3,.4] & - & [.7,.9] \\ [.1,.2] & [.2,.4] & [.1,.3] & - \end{pmatrix}.$$

9.4.1 First Round

Once the experts have provided their preferences, the CSS carries out the first round of the consensus reaching process following the phases described in the Sect. 9.3.

Making the Information Uniform

In this phase the heterogeneous information is unified into a common domain S_T. As we said in the Sect. 9.3.1, once an appropriate S_T has been chosen, the model applies different transformation functions τ_{DS_T} to transform each expert's preference into a fuzzy set defined on S_T, obtaining the following fuzzy sets:

$$\tilde{\mathbf{P}}_{\mathbf{e_1}} = \begin{pmatrix} - & (0,0,0,1,0,0,0) & (0,0,0,0,.19,.81,0) & (0,0,.59,.41,0,0,0) \\ (0,.19,.81,0,0,0,0) & - & (0,0,0,0,0,.59,.41) & (0,.19,.81,0,0,0,0) \\ (0,.19,.81,0,0,0,0) & (0,.81,.19,0,0,0,0) & - & (0,0,.59,.41,0,0,0) \\ (0,0,0,0,0,.59,.41) & (0,0,0,0,.19,.81,0) & (0,0,0,1,0,0,0) & - \end{pmatrix}$$

$$\tilde{\mathbf{P}}_{\mathbf{e_2}} = \begin{pmatrix} - & (0,0,0,0,1,0,0) & (0,0,0,0,0,1,0) & (0,0,0,1,0,0,0) \\ (0,0,1,0,0,0,0) & - & (0,0,0,0,1,0,0) & (0,0,0,0,0,1,0) \\ (0,1,0,0,0,0,0) & (1,0,0,0,0,0,0) & - & (0,0,0,0,0,1,0) \\ (0,0,1,0,0,0,0) & (0,1,0,0,0,0,0) & (1,0,0,0,0,0,0) & - \end{pmatrix}$$

$$\tilde{\mathbf{P}}_{\mathbf{e_3}} = \begin{pmatrix} - & (0,0,0,0,.81,.81,0) & (0,0,0,.12,1,.19,0) & (0,0,0,0,.19,1,.41) \\ (0,.19,1,.12,0,0,0) & - & (0,0,0,.41,1,.19,0) & (0,0,0,0,.19,1,.12) \\ (0,.19,1,.12,0,0,0) & (0,.19,1,.41,0,0,0) & - & (0,0,0,0,.81,1,.41) \\ (.41,1,.19,0,0,0,0) & (0,.81,1,.41,0,0,0) & (.41,1,.81,0,0,0,0) & - \end{pmatrix}$$

Computing Consensus Degrees

1. *Central values.* Applying (9.2), the model computes the central values of the fuzzy sets:

$$cv(e_1) = \begin{pmatrix} - & 3 & 4.81 & 2.41 \\ 1.81 & - & 5.41 & 1.81 \\ 1.81 & 1.19 & - & 2.41 \\ 5.41 & 4.81 & 3 & - \end{pmatrix} \quad cv(e_2) = \begin{pmatrix} - & 4 & 5 & 3 \\ 2 & - & 4 & 5 \\ 1 & 0 & - & 5 \\ 2 & 1 & 0 & - \end{pmatrix}$$

$$cv(e_3) = \begin{pmatrix} - & 4.5 & 4 & 5.13 \\ 1.94 & - & 3.86 & 4.94 \\ 1.94 & 2.13 & - & 4.81 \\ 0.86 & 1.81 & 1.18 & - \end{pmatrix}$$

2. *Similarity matrices.* The model computes a similarity matrix between each pair of experts by using the distance function (9.1):

$$SM_{12} = \begin{pmatrix} - & 0.83 & 0.96 & 0.9 \\ 0.96 & - & 0.76 & 0.46 \\ 0.86 & 0.8 & - & 0.56 \\ 0.43 & 0.36 & 0.5 & - \end{pmatrix} \quad SM_{13} = \begin{pmatrix} - & 0.75 & 0.87 & 0.54 \\ 0.97 & - & 0.74 & 0.47 \\ 0.97 & 0.84 & - & 0.59 \\ 0.24 & 0.5 & 0.69 & - \end{pmatrix}$$

$$SM_{23} = \begin{pmatrix} - & 0.91 & 0.84 & 0.64 \\ 0.99 & - & 0.97 & 0.99 \\ 0.84 & 0.64 & - & 0.97 \\ 0.81 & 0.86 & 0.8 & - \end{pmatrix}$$

3. *Consensus matrix.* The model calculates the consensus matrix by aggregating the similarity matrices:

$$CM = \begin{pmatrix} - & 0.83 & 0.89 & 0.69 \\ 0.97 & - & 0.82 & 0.64 \\ 0.89 & 0.76 & - & 0.71 \\ 0.49 & 0.57 & 0.66 & - \end{pmatrix}$$

4. *Consensus degrees.* The model computes the consensus degree at different levels:
   Level 1. *Consensus on pairs of alternatives.* The element (l, k) of CM represents the consensus degree on the pair of alternatives (x_l, x_k).
   Level 2. *Consensus on alternatives.*

$$ca^1 = 0.8,\ ca^2 = 0.81,\ ca^3 = 0.78,\ ca^4 = 0.57$$

   Level 3. *Consensus on the relations or global consensus.*

$$cr = 0.74$$

From these results, we can draw some conclusions:

1. The level of agreement in the pair (21) is very high, $cp^{21} = 0.97$, it means that the assessments given on that pair are very similar. On the contrary, the assessments given on the pair (41) have to be enough different because $cp^{41} = 0.49$ is low.
2. The alternative where the agreement is bigger is x_2, while the alternative with smaller agreement is x_4.
3. The level of global agreement among experts is not bad, $cr = 0.74$, but as we shall see following, it is not enough to finish the consensus process.

Checking the Agreement

In this phase the global consensus value cr is compared with the consensus threshold γ. In this example, we have decided to use a high consensus threshold, $\gamma = 0.8$. As $cr = 0.74 < \gamma$, the current consensus is not big enough to finish the consensus process and therefore the process must continue.

Production of Advice

In this phase the CSS identifies what preferences should be changed and how to carry out these changes.

Computation of Proximity Measures

The model computes the collective preference relation aggregating all individual preference relations using the average as aggregation operator:

1. Computing collective preferences

$$\begin{aligned}
p_c^{12} &= (0, 0, 0, 0.33, 0.6, 0.27, 0)\\
p_c^{13} &= (0, 0, 0, 0.4, 0.39, 0.66, 0)\\
p_c^{14} &= (0, 0, 0.19, 0.47, 0.06, 0.33, 0.13)\\
p_c^{21} &= (0, 0.12, 0.93, 0.04, 0, 0, 0)\\
p_c^{23} &= (0, 0, 0, 0.13, 0.66, 0.26, 0.13)\\
p_c^{24} &= (0, 0.06, 0.27, 0, 0.06, 0.66, 0.04)
\end{aligned}$$

$$p_c^{31} = (0, 0.46, 0.6, 0.04, 0, 0, 0)$$
$$p_c^{32} = (0.33, 0.33, 0.39, 0.13, 0, 0, 0)$$
$$p_c^{34} = (0, 0, 0.19, 0.13, 0.27, 0.66, 0.13)$$
$$p_c^{41} = (0.13, 0.33, 0.39, 0, 0, 0.19, 0.13)$$
$$p_c^{42} = (0, 0.6, 0.33, 0.13, 0.06, 0.27, 0)$$
$$p_c^{43} = (0.47, 0.33, 0.27, 0.33, 0, 0, 0)$$

2. Proximity matrices. A proximity matrix for each expert is obtained:

$$PM_1 = \begin{pmatrix} - & 0.84 & 0.95 & 0.77 \\ 0.98 & - & 0.82 & 0.63 \\ 0.96 & 0.98 & - & 0.68 \\ 0.5 & 0.58 & 0.72 & - \end{pmatrix}; \quad PM_2 = \begin{pmatrix} - & 0.99 & 0.92 & 0.86 \\ 0.98 & - & 0.94 & 0.83 \\ 0.89 & 0.78 & - & 0.88 \\ 0.92 & 0.77 & 0.77 & - \end{pmatrix}$$

$$PM_3 = \begin{pmatrix} - & 0.9 & 0.91 & 0.77 \\ 0.99 & - & 0.92 & 0.84 \\ 0.94 & 0.85 & - & 0.91 \\ 0.73 & 0.91 & 0.97 & - \end{pmatrix}$$

3. Proximity measures. The model computes the proximity at different levels:
   Level 1. *Proximity on pairs of alternatives.* These values are equal to values of the proximity matrices.
   Level 2. *Proximity on alternatives.*

x_1	x_2	x_3	x_4
$pa_1^1 = 0.85$	$pa_1^2 = 0.81$	$pa_1^3 = 0.87$	$pa_1^4 = 0.6$
$pa_2^1 = 0.92$	$pa_2^2 = 0.92$	$pa_2^3 = 0.85$	$pa_2^4 = 0.82$
$pa_3^1 = 0.86$	$pa_3^2 = 0.92$	$pa_3^3 = 0.9$	$pa_3^4 = 0.87$

   Level 3. *Proximity on the relation.*

$$pr_1 = 0.78, pr_2 = 0.88, pr_3 = 0.89$$

According to the results, the furthest expert is e_1 and the nearest expert is e_3.

Guided Advice Generator

The model applies the identification rules to identify what preferences have to be changed and the direction rules to suggest how to make the changes.

Identification Rules

1. Set of experts to change their preferences, $EXPCH$. The ranking of the experts according to their proximity is e_3, e_2, e_1, being e_1 the furthest expert. In this example, like we are working with three experts, we will suggest that only one change their assessments, i.e., $ne = 1$:

$$EXPCH = \{e_1\}.$$

2. Set of alternatives whose assessments should be changed, ALT. In this case, as we have fixed a consensus threshold $\gamma = 0.8$, we have:

$$ALT = \{x_l \in X |\ ca^l < 0.8\} = \{x_3, x_4\}.$$

3. Set of pairs of alternatives whose associated assessments should be changed, $PALT_i$. At this point, the model identifies the pairs of alternatives that have to be changed taking into account a proximity threshold $\beta = 0.75$:

$$PALT_1 = \{(x_3, x_4), (x_4, x_1), (x_4, x_2), (x_4, x_3)\}$$

Finally, the list of preference to change is:

$$p_1^{34}, p_1^{41}, p_1^{42}, p_1^{43}$$

Direction Rules

1. Direction parameters.

	$\big(p_i^{lk}(ml_{pos}), p_i^{lk}(ml_{val}), p_i^{lk}(sl_{pos}), p_i^{lk}(sl_{val})\big)$	$\big(p_c^{lk}(ml_{pos}), p_c^{lk}(ml_{val}), p_c^{lk}(sl_{pos}), p_c^{lk}(sl_{val})\big)$
p_1^{34}	$(2, 0.59, 3, 0.41)$	$(5, 0.66, 4, 0.27)$
p_1^{41}	$(5, 0.59, 6, 0.41)$	$(2, 0.39, 1, 0.33)$
p_1^{42}	$(5, 0.81, 4, 0.19)$	$(1, 0.6, 2, 0.33)$
p_1^{43}	$(3, 1, *, 0)$	$(0, 0.47, 2, 0.27)$

(*) means that there are more than one possible secondary label candidates but they do not play any role in the production of recommendations.

2. Application of the direction rules.
   - Given that $p_1^{41}(ml_{pos}) > p_c^{41}(ml_{pos})$, $p_1^{42}(ml_{pos}) > p_c^{42}(ml_{pos})$ and $p_1^{43}(ml_{pos}) > p_c^{43}(ml_{pos})$, expert e_1 is advised to decrease these assessments according to the rule DR1.
   - Given that $p_1^{34}(ml_{pos}) < p_c^{41}(ml_{pos})$ expert e_1 is advised to increase this assessment according to the rule DR2.

9.4.2 Second Round

Following the previous advice given by de CSS model, the expert e_1 changes his preferences. In order to avoid abrupt changes in experts' preferences, we have decided to increase or decrease the current assessments 0.2.

$$\mathbf{P_{e_1}} = \begin{pmatrix} - & .5 & .8 & .4 \\ .3 & - & .9 & .3 \\ .3 & .2 & - & \mathbf{.6} \\ \mathbf{.7} & \mathbf{.6} & \mathbf{.3} & - \end{pmatrix}$$

Due to the CSS carries out the same operations in all rounds of consensus, in the following rounds we only show the results that provide us bigger information about the performance of the model.

Making the Information Uniform

The operations in this phase are the same than in the first round.

Computing Consensus Degree

1. *Consensus matrix.*

$$CM = \begin{pmatrix} - & 0.83 & 0.89 & 0.69 \\ 0.97 & - & 0.82 & 0.64 \\ 0.89 & 0.76 & - & 0.84 \\ 0.56 & 0.71 & 0.79 & - \end{pmatrix}$$

2. *Consensus degrees.* The model computes the consensus degree at different levels:

   Level 1. *Consensus on pairs of alternatives.* Elements (l, k) of the consensus matrix CM.

   Level 2. *Consensus on alternatives.*

$$ca^1 = 0.8,\ ca^2 = 0.81,\ ca^3 = 0.83,\ ca^4 = 0.69$$

   Level 3. *Consensus on the relations or global consensus.*

$$cr = 0.78$$

By comparing the results obtained in the first and second round, we can highlight that:

1. The level of agreement in the pair (41), $cp^{41} = 0.56$, is bigger in the second round than in the first round, $cp^{41} = 0.49$, therefore we can verify that decreasing the value given by the expert e_1 on p_1^{41}, e_1 has been able to bring near his assessment to the assessments given by e_2 and e_3.
2. The level of agreement in the alternatives affected by the changes has increased, therefore the correct direction of the changes have been recommended.

Checking the Agreement

Given that $cr = 0.78 < \gamma = 0.8$, the consensus degree is not big enough yet and the consensus process must continue.

Production of Advice

Computation of Proximity Measure

1. *Proximity measures.* The model computes the proximity at different levels:
   Level 1. *Proximity on pairs of alternatives* for expert e_i are given in PM_i.
   Level 2. *Proximity on alternatives.*

x_1	x_2	x_3	x_4
$pa_1^1 = 0.85$	$pa_1^2 = 0.81$	$pa_1^3 = 0.92$	$pa_1^4 = 0.73$
$pa_2^1 = 0.92$	$pa_2^2 = 0.92$	$pa_2^3 = 0.87$	$pa_2^4 = 0.86$
$pa_3^1 = 0.86$	$pa_3^2 = 0.92$	$pa_3^3 = 0.92$	$pa_3^4 = 0.9$

   Level 3. *Proximity on the relation.*

$$pr_1 = 0.83, pr_2 = 0.89, pr_3 = 0.9$$

Note that although e_1 has been able to bring near his preferences to the collective preference in the second round (from $pr_1 = 0.78$ to $pr_1 = 0.83$), e_1 continues being the furthest experts, and therefore, the CSS model will recommend him to change his preferences again.

Guided Advice Generator

Identification Rules

1. Set of experts to change their preferences, $EXPCH$.

$$EXPCH = \{e_1\}$$

2. Set of alternatives whose assessments should be changed, ALT.

$$ALT = \{x_l \in X |\ ca^l < 0.8\} = \{x_4\}$$

3. Set of pairs of alternatives whose associated assessments should be changed, $PALT_i$.

$$PALT_1 = \{(x_4, x_1), (x_4, x_2)\}$$

   List of preference to change:

$$p_1^{41}, p_1^{42}$$

Direction Rules

1. Direction parameters.

	$(p_i^{lk}(ml_{pos}), p_i^{lk}(ml_{val}), p_i^{lk}(sl_{pos}), p_i^{lk}(sl_{val}))$	$(p_c^{lk}(ml_{pos}), p_c^{lk}(ml_{val}), p_c^{lk}(sl_{pos}), p_c^{lk}(sl_{val}))$
p_1^{41}	$(5, 0.81, 4, 0.19)$	$(2, 0.39, 1, 0.33)$
p_1^{42}	$(4, 0.59, 3, 0.41)$	$(1, 0.6, 2, 0.33)$

2. Application of the direction rules.
   - Due to fact that $p_1^{41}(ml_{pos}) > p_c^{41}(ml_{pos})$ and $p_1^{42}(ml_{pos}) > p_c^{42}(ml_{pos})$, expert e_1 is advised to decrease these assessments according to the rule DR1.

9.4.3 Third Round

Following the advice given in the second round by de CSS, the expert e_1 changes his preferences.

$$\mathbf{P_{e_1}} = \begin{pmatrix} - & .5 & .8 & .4 \\ .3 & - & .9 & .3 \\ .3 & .2 & - & .6 \\ \mathbf{.5} & \mathbf{.4} & .3 & - \end{pmatrix}$$

Making the Information Uniform

The operations in this phase are the same than in the first round.

Computing Consensus Degree

1. *Consensus matrix.*

$$CM = \begin{pmatrix} - & 0.83 & 0.89 & 0.69 \\ 0.97 & - & 0.82 & 0.64 \\ 0.89 & 0.76 & - & 0.84 \\ 0.76 & 0.84 & 0.79 & - \end{pmatrix}$$

2. *Consensus degrees.* The model computes the consensus degree at different levels:

   Level 1. *Consensus on pairs of alternatives.* Elements (l, k) of the consensus matrix CM.

   Level 2. *Consensus on alternatives.*

$$ca^1 = 0.8,\ ca^2 = 0.81,\ ca^3 = 0.83,\ ca^4 = 0.8$$

   Level 3. *Consensus on the relations or global consensus.*

$$cr = 0.81$$

Checking the Agreement

Finally, in the third round the level of agreement is bigger than the consensus threshold, $cr = 81 > \gamma = 0.8$. Therefore, the experts have been able to reach the minimum level of agreement fixed initially and the consensus reaching process should finish. Immediately afterward, a selection process should be run to obtain the final solution of the decision problem.

As summary of this section and according to the results shown in each consensus rounds, if the experts follow the recommendation given by the model, we can affirm that the CSS achieves to increase the level of agreement during the consensus reaching process.

9.5 Conclusions

In this chapter we have proposed a CSS model to automate the consensus processes in GDM problems where the experts use different information domain to provide their opinions. Two main features may be emphasized about this model: (1) it is able to manage consensus processes in problems where experts use numerical, interval-valued or linguistic assessment to express their preferences, and (2) it is able to suggest the changes of preferences that experts should apply in order to reach the wanted consensus. The model can be used to substitute the figure of the moderator, avoiding in this way a possible moderator's partiality during the consensus reaching process.

This CSS model uses a methodology based on transformation functions to unify the heterogeneous information into a common domain. It also defines a similarity function based on central values of the fuzzy sets to compute two kind of measurements: the consensus degree and the proximity values. These calculations are carried out at three different levels: pairs of alternatives, alternatives and relations. Based on both measurements, a guided advice system has been designed to help the experts to identify the preferences where the disagreement is bigger and to suggest how to change such preferences in order to increase the agreement among the experts.

Once the experts have changed their preferences and have achieved a high level of consensus, they are prepared to carry out the process to choose the best alternative(s) to solve the outlined problem.

Acknowledgements

This chapter has been supported by the Research Project TIN2006-02121 and FEDER funds.

References

P.P. Bonissone and K.S. Decker. Selecting uncertainty calculi and granularity: An experiment in trading-off precision and complexity. *Uncertainty in Artificial Intelligence*, pages 217–247. North-Holland, Amsterdam, 1986.

G. Bordogna, M. Fedrizzi, and G. Pasi. A linguistic modeling of consensus in group decision making based on OWA operators. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 27:126–132, 1997.

N. Bryson. Group decision-making and the analytic hierarchy process: Exploring the consensus-relevant information content. *Computers and Operational Research*, 1(23):27–35, 1996.

T.X. Bui. *A Group Decison Support System for Cooperative Multiple Criteria Group Decison-Making*. Springer, Berlin Heidelberg New York, 1987.

C. Carlsson, D. Ehrenberg, P. Eklund, M. Fedrizzi, P. Gustafsson, P. Lindholm, G. Merkuryeva, T. Riissanen, and A.G.S. Ventre. Consensus in distributed soft environments. *European Journal of Operational Research*, 61:165–185, 1992.

M. Delgado, F. Herrera, E. Herrera-Viedma, and L. Martínez. Combinig numerical and linguistic information in group decision making. *Information Sciences*, 107:177–194, 1998.

Z.P. Fan and X. Chen. Consensus measures and adjusting inconsistency of linguistic preference relations in group decision making. *Lecture Notes in Artificial Intelligence*, 3613:130–139, 2005.

J. Fodor and M. Roubens. *Fuzzy Preference Modelling and Multicriteria Decision Support*. Kluwer, Dordrecht, 1994.

F. Herrera and E. Herrera-Viedma. Linguistic decision analysis: Steps for solving decision problems under linguistic information. *Fuzzy Sets and Systems*, 115:67–82, 2000.

F. Herrera and L. Martínez. An approach for combining linguistic and numerical information based on 2-tuple fuzzy representation model in decision-making. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 8(5):539–562, 2000.

F. Herrera, E. Herrera-Viedma, and J.L. Verdegay. A model of consensus in group decision making under linguistic assessments. *Fuzzy Sets and Systems*, 79:73–87, 1996.

F. Herrera, L. Martínez, and P.J. Sánchez. Managing non-homogeneous information in group decision making. *European Journal of Operational Research*, 166(1):115–132, 2005.

E. Herrera-Viedma, F. Herrera, and F. Chiclana. A consensus model for multiperson decision making with different preference structures. *IEEE Transactions on Systems, Man and Cybernetics-Part A: Systems and Humans*, 32(3):394–402, 2002.

E. Herrera-Viedma, F. Mata, L. Martínez, F. Chiclana, and L.G. Pérez. Measurements of consensus in multi-granular linguistic group decision making. *Lecture Notes in Artificial Intelligence*, 3131:194–204, 2004.

E. Herrera-Viedma, L. Martínez, F. Mata, and F. Chiclana. A consensus support system model for group decision-making problems with multi-granular linguistic preference relations. *IEEE Transactions on Fuzzy Systems*, 13(5):644–658, 2005.

J. Kacprzyk. Group decision making with a fuzzy linguistic majority. *Fuzzy Sets and Systems*, 18:105–118, 1986.

J. Kacprzyk, M. Fedrizzi, and H. Nurmi. “Soft” degrees of consensus under fuzzy preferences and fuzzy majorities. *Consensus Under Fuzziness*, pages 55–81. Kluwer, Dordrecht, 1997.

S.H. Kim, S.H. Choi, and J.K. Kim. An interactive procedure for multiple attribute group decision making with incomplete information: Range-based approach. *European Journal of Operational Research*, 118:139–152, 1999.

D. Kuchta. Fuzzy capital budgeting. *Fuzzy Sets and Systems*, 111:367–385, 2000.

L.I. Kuncheva. Five measures of consensus in group decision making using fuzzy sets. In *International Conference on Fuzzy Sets and Applications IFSA-95*, pages 141–144, 1991.

S. Kundu. Min-transitivity of fuzzy leftness relationship and its application to decision making. *Fuzzy Sets and Systems*, 86:357–367, 1997.

J.F. Le Téno and B. Mareschal. An interval version of PROMETHEE for the comparison of building products' design with ill-defined data on environmental quality. *European Journal of Operational Research*, 109:522–529, 1998.

E. Levrat, A. Voisin, S. Bombardier, and J. Bremont. Subjective evaluation of car seat comfort with fuzzy set techniques. *International Journal of Intelligent Systems*, 12:891–913, 1997.

M. Roubens. Fuzzy sets and decision analysis. *Fuzzy Sets and Systems*, 90:199–206, 1997.

S. Saint and J.R. Lawson. *Rules for Reaching Consensus. A Modern Approach to Decision Making.* Jossey-Bass, San Francisco, 1994.

E. Szmidt and J. Kacprzyk. A consensus reaching process under intuitionistic fuzzy preference relations. *International Journal of Intelligent System*, 18(7):837–852, 2003.

R.R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Transactions on Systems, Man and Cybernetics*, 18:183–190, 1988.

R.R. Yager. An approach to ordinal decision making. *International Journal of Approximate Reasoning*, 12:237–261, 1995.

R.R. Yager. Protocol for negotiations among multiple intelligent agents. *Consensus Under Fuzziness*, pages 165–174. Kluwer, Dordrecht, 1997.

L.A. Zadeh. The concept of a linguistic variable and its applications to approximate reasoning. *Information Sciences, Part I, II, III*, 8, 8, 9:199–249, 301–357, 43–80, 1975.

L.A. Zadeh. Fuzzy logic = computing with words. *IEEE Transactions on Fuzzy Systems*, 4(2):103–111, 1996.

S. Zadrozny. An approach to the consensus reaching support in fuzzy environment. *Consensus Under Fuzziness*, pages 83–109. Kluwer, Dordrecht, 1997.

Q. Zhang, J.C.H. Chen, and P.P. Chong. Decision consolidation: Criteria weight determination using multiple preference formats. *Decision Support Systems*, 38(2):247–258, 2004.

10

Evaluating Medical Decision Making Heuristics and Other Business Heuristics with Neural Networks

Steven Walczak

The Business School, University of Colorado at Denver and Health Sciences Center, Campus Box 165, PO Box 173364, Denver, CO 80217 USA
Steven.Walczak@cudenver.edu.

Summary. Heuristics are an efficient means for solving complex and also partial information business problems. Unfortunately, the development of new heuristics and the evaluation of existing heuristics is a labor intensive process. Neural networks provide a fast and reliable method for evaluation of new heuristics against existing heuristics and the optimization of new heuristics when no prior heuristic exists. This chapter describes a methodology for utilizing neural networks as a heuristic evaluation mechanism and discusses how existing research has been utilized (possibly unintentionally) in the development or evaluation of new heuristics.

10.1 Introduction

Businesses face difficult problems on a daily basis. Historically, heuristic methods have been used to solve complex and nonlinear business problem types, such as frequently occur in medicine, finance, and other business areas. While heuristics provide an agile means to quickly evaluate complex independent variables to make a decision for a specific business problem, heuristics by their nature may not always provide an optimal solution (Pearl 1984).

For the purposes of this chapter, a heuristic is defined as any "rule of thumb", that is a decision making model, that enables rapid decision making in domains where complete information regarding the decision making problem is either very complex and thus difficult to evaluate or may be missing. A heuristic is therefore a decision making rule of some type. Elements of heuristics are the information, variables, that are available and are evaluated to form the heuristic decision. Hence, while the variables themselves are not heuristics, they are a necessary criteria for employing a heuristic decision making method and will be treated as equivalent to the heuristic method itself in this chapter.

An interesting question then is how businesses and services may evaluate current and new heuristics or heuristic variables to determine their

S. Walczak: *Evaluating Medical Decision Making Heuristics and Other Business Heuristics with Neural Networks*, Studies in Computational Intelligence (SCI) **97**, 259–287 (2008)
www.springerlink.com

ability to select optimal or near optimal solutions. Of course, proving that a heuristic is optimal is a much harder problem, but evaluation techniques are meant to build confidence in the reliability of the heuristic being evaluated and not to prove optimality. Traditional statistics provide a number of tools for performing such evaluations (e.g., mean error rate, regression modeling, and discriminant analysis), but these evaluation methods suffer from rigid a priori requirements on population distributions and error distributions (Daniel 1999), which frequently are unknown for traditional medical and other business problems.

When a priori error, population, and variable distributions are unknown, nonparametric methods need to be applied (McLachlan 1992). Neural networks are a nonparametric methodology, which means that they model underlying equations without any requirement for a priori knowledge of population or variable distributions.

Although neural networks are often described as a “black box”, indicating that information concerning the effect of individual variables on predicted outcomes is difficult to ascertain, they have several advantages that make them an ideal choice in evaluating (as well as developing) decision making models in medicine and business. These advantages are:

- Extremely fast, once trained
- Tolerant of moderate amounts of noise in data
- Prediction and classification models are learned from the data dynamically, producing optimal or near-optimal models

While the idea of a heuristic model is to use a smaller and possibly less expensive set of variables to produce accurate decisions, another advantage of neural networks is in their ability to handle not only noisy data, but also missing data for the heuristic model’s variables (Markey et al. 2006).

The possible difficulty of determining the effect of individual decision variables for heuristics presents a problem, this problem may be overcome in several ways as will be discussed in this chapter. The focus of this chapter is on the utilization of neural networks as a tool for evaluating either new heuristic decision models or more frequently evaluating the utility of new or different combinations of variables to either create a new heuristic model or improve upon the performance of an existing heuristic decision model.

The remainder of the chapter is organized as follows, First, factors that impact the performance of neural networks and values for neural network model building are examined. Next, a process for utilizing neural networks to evaluate new heuristic decision making models, including new combinations of variables that serve as the input to the heuristic method, is presented and discussed. Finally, historic examples of the use of neural networks to discover new heuristics or invalidate existing heuristics are presented for both medical and business domains.

10.2 Neural Network Parameters

The advantages of neural networks just mentioned appear to make them an ideal choice for both developing heuristic decision models as well as evaluating existing and new heuristic decision models.

10.2.1 Training Time and "Black Box" Nature

A disadvantage noted in the literature which is no longer true is that even though neural networks are extremely fast, even instantaneous, once trained, the training time can require a large time investment. Current neural network shell programs that assist researchers and modelers in developing neural network solutions typically require only a few minutes to train a moderate size network (less than 100 processing nodes and 2,500 connections). As such the training time cost is negligible for most heuristic decision making problems, since the utilization of heuristics simplifies the decision space. Of course as the size of the neural network architecture grows, the training time requirement will increase correspondingly.

The other significant disadvantage of utilizing neural networks for evaluating the applicability and effectiveness of business heuristics is the "black box" nature of neural networks. This is a disadvantage in evaluating heuristics, since the effect of the variables belonging to the heuristic technique may be difficult to determine. Section 2.3 will discuss a means for overcoming this handicap, but right now the ability to produce an algorithm where independent variables are defined is examined. The "black-box" nature of neural networks is being challenged and various techniques are being developed to mine the variable relationships that drive neural network performance (Zhang 2007) and produce if-then decision heuristics from neural network connections (Ballini and Gomide 2002/2003).

The small neural network shown in Fig. 10.1, utilizes two independent input variables, x_1 and x_2 and produces either a prediction or classification value y. An aggregation algorithm specified by the neural network developer is

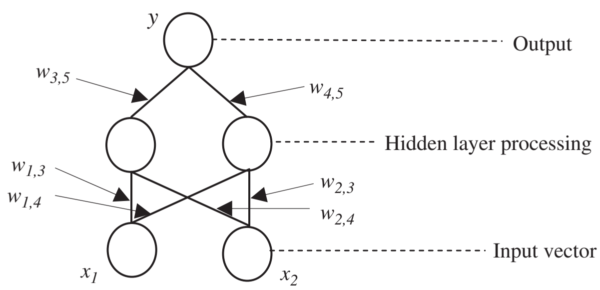


Fig. 10.1. Sample neural network

used to combine input values into a processing node, $f(x_1,\ x_2)$, and a transfer function, $g(X)$, applies an activation function for transferring the nodes value to the next layer in the neural network. Thus:

$$\bullet\ y = f_5(g_3(f_3(x_1, x_2), g_4(f_4(x_1, x_2)), \tag{10.1}$$

where the subscripts indicate the aggregation or transfer function of a particular processing or output node in the neural network. If a simple weighted summation is used for all $f(x_1,\ x_2)$, and a sigmoid is used for all $g(X)$, then (1) becomes:

$$\bullet\ y = \left(\sum_{i=3}^{4} w_{i,5} \frac{1}{1 + e^{-\sum_{m=1}^{2} w_{m,j} x_m}} \right) + \varepsilon. \tag{10.2}$$

The range of i in (2) is the number of hidden processing nodes and the range of m is the number of input nodes. This equation is for a single hidden layer. Additional hidden layers would mean aggregating the weighted sum of the sigmoid function applied to separate copies of (2) for each additional layer processing node. As can be seen, while it is possible to generate an equation and determine variable impact, the size of the equation quickly becomes problematic for understanding the relevance of an individual variable (or collection of variables) as additional processing nodes and hidden layers are added.

An alternate means to evaluate the impact of input variables on the model's output is to examine the values for each connecting weight, $w_{i,j}$. The backpropagation algorithm learns to model domain problems by automatically adjusting these weights to approximate an optimal model, hence neural networks may themselves be seen as heuristic models. Every connection weight in the model must be examined to try and determine the overall influence of specific variables, with the absolute value of the weight serving as an indicator. This technique, like the algorithm generation technique just described is very difficult to accomplish and becomes more problematic as the size of the neural network's hidden node architecture increases.

One final technique and one that is commonly performed by existing neural network shell tools and that is easier to perform is the leave-one-out methodology. This technique emulates the step-wise portion of step-wise regression modeling. Leave-one-out, as the name describes, involves dropping or alternately adding a single variable and comparing the performance of the two corresponding neural network models that differ by only the single variable. If a significant increase or decrease in performance is noted, then the effect of the missing or added variable may be estimated. A drawback of this process is that other interaction effects are likely to be occurring in the neural network and so the final difference in model results is not solely due to the single left out variable. This technique will form a part of the neural network heuristic evaluation process described in the next methodology section.

10.2.2 Other Factors Affecting Neural Network Efficacy for Heuristic Evaluation

Two other factors affecting neural network model development that are debated in the research community, which may turn out to be advantages for using a neural network based nonparametric modeling approach are: training population size and input vector size.

Typically, especially in financial and medical domains, researchers prefer to have larger sample populations for developing and validating decision models (Zahedi 1996; Zhang and Hu 1998). Larger data populations improve statistical significance of the resulting model by increasing the probability that unusual cases are included in the sample population (Schürmann 1996). Recent research has indicated that, at least for the populations studied, much smaller training populations still enable neural network models to achieve very high prediction and classification accuracy rates (Abdel-Aal 2004; Shin et al. 2005; Walczak 2001). Based on these reported research results, it may be possible to utilize neural networks for heuristic model evaluation with much smaller sample (training) sizes than are required for validation utilizing traditional statistical methods.

Even with a very small population of historic data, which is needed for the supervised learning methods, reasonable approximations of the performance of neural network heuristic models may be accurately evaluated utilizing several techniques. Bootstrapping and the jackknifing refinement of bootstrapping (Efron 1982) enable the utilization of members in the historic population set to be utilized as both training and evaluation cases through utilizing a leave one out methodology (Tourassi and Floyd 1997). Similarly, cross-fold (commonly called N-fold) validation is a similar method to jackknifing but which utilizes larger sets of randomly selected evaluation cases from the population and is commonly used in algorithmic validation experiments (Lim et al. 2000).

The other design issue debated among neural network researchers is the relative advantage or disadvantage of learning with respect to input vector size. Early researchers claimed that an advantage of the fact that neural networks learned connection weights was that large quantities of independent variables could be included in a neural network model's input vector and that the neural network learning algorithm would deselect noncontributing variables out of the model (see e.g., Hertz et al. 1991). This could be advantageous, if it works, in that variable additions to heuristics could be tested by adding them in and determining if the neural network has deselected any of the new heuristic variables. Alternatively, if the neural network keeps a heuristic variable over more traditional variable (through the process of deselection) this could serve as validation of the new heuristic.

However, the other side of the input vector size argument claims that each input vector variable impacts the resulting classification or prediction made by the neural network and hence the selection of input variables to the neural network model is a critical decision factor and determines the optimality of the corresponding neural network model (Güler and Übeyli 2005; Nath et al. 1997; Piramuthu et al. 1994; Smith 1993; Soulié 1994; Tahai et al. 1998; Weigand and Zimmermann 1995). In fact, Pakath and Zaveri (1995) claim that input variable selection sensitivity is not only a factor for neural networks, but for other artificial intelligence modeling paradigms as well. This too can serve in the goal of heuristic evaluation, but does remove the simplicity of automation.

The author of this chapter follows the later design criteria, which typically involves the use of domain experts to determine if potential variables could contribute to a heuristic solution to the problem being modeled. One consideration is that highly correlated variables must be removed, so that only the influence from a single variable (or set) affects the neural network model's outcome (Smith 1993), which involves some additional up front analysis of the variables being used.

Additional design considerations must be determined by the researcher desiring to evaluate decision heuristics, such as the type of training algorithm used and the architecture of the neural network model, but will not be discussed at length here as they have already been discussed in detail in the literature (for tutorials on neural network implementation and design issues see (Jain et al. 1996; Rodvold et al. 2001; Zhang 2007)). The remainder of this chapter will focus on supervised learning methods for neural network training, including the very popular backpropagation algorithm, though the techniques described are applicable to unsupervised learning methodologies as well.

10.3 Methodology for Evaluating Business/Medical Heuristics

The advantages of a learning system that dynamically determines classification or prediction equations, is noise tolerant, and may be able to be used with relatively small amounts of data provides an interesting potential for evaluating the potential of business heuristics. As noted in Sect. 2.3 though, determining the contribution of specific input vector variables is problematic and researchers are advised to limit the quantity of input vector variables to be able to more critically evaluate the effect of the heuristic model's variables.

Since input vector variable selection is viewed as a critical step in the development of neural network models, the heuristic variables may be evaluated by adding them to a current set of decision variables and possibly deleting correlated variables (which may be more difficult or more costly to acquire) and evaluating if any classification or prediction performance improvement has occurred, similar to the leave-one-out method described earlier. Typically, this methodology assumes that an existing decision model already exists. If no existing method is available for comparison, then it simply becomes a case of evaluating a new model, regardless of the heuristic nature, and confidence values for the new models output should suffice.

When a competing model is already in use, then two neural network models are required. Since typically a new heuristic decision model will utilize a different number of input variables, the architecture for each model must be optimized independently, following traditional neural network development protocol. Optimizing the number of hidden nodes in each layer and the number of layers is critical for comparing optimal results for each method. The methodology for comparing a new heuristic model is displayed in flowchart format in Fig. 10.2.

The two critical parts of the flowchart in Fig. 10.2 are in the two decision diamonds. Selecting an appropriate statistical method for comparing the output of the two models is important since classification models and forecasting models require different analysis methods and selecting an inappropriate

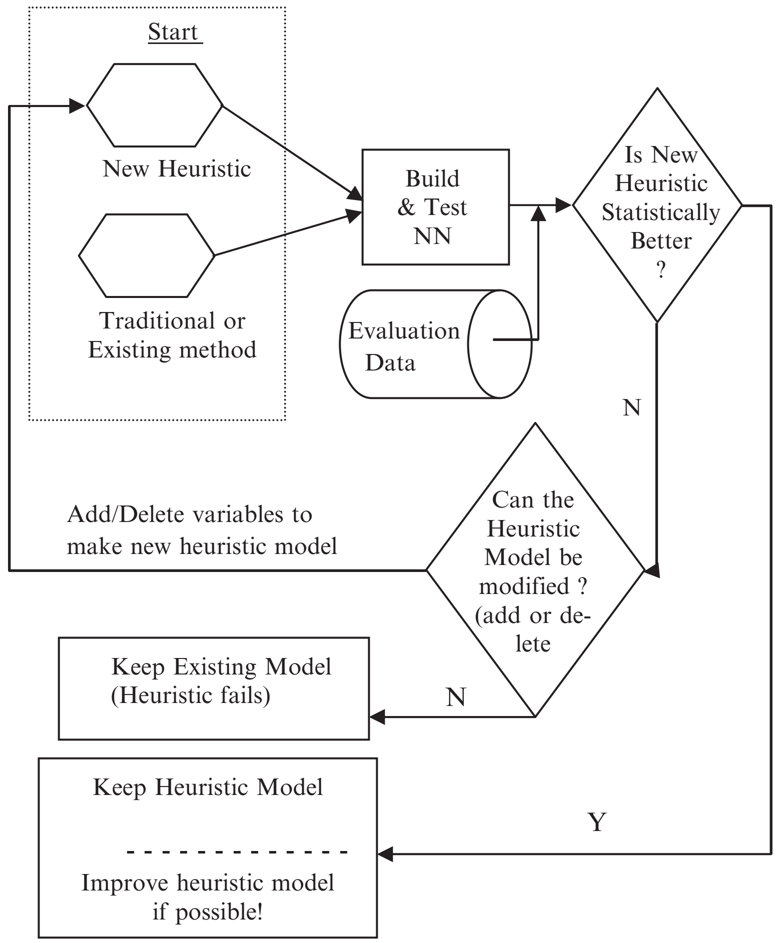


Fig. 10.2. Flowchart of new heuristic comparison methodology

statistic for comparison could significantly alter the evaluation results. In heuristic medical models that involve patients, the sensitivity and specificity of the model's output is the gold standard for comparison, however trade-offs in sensitivity and specificity may be possible. Furthermore, for medical domains, it has been recommend that all neural network models be evaluated using clinical trials or randomized control trials of the neural network's model of the heuristic (Lisboa and Taktak 2006) to demonstrate real world performance of the heuristic model.

The other important aspect of heuristic development is the ability to modify the heuristic through either the inclusion of new decision variables or the exclusion of current variables in the heuristic model. Since the goal of a heuristic is to simplify the decision making task in a complex decision environment, then improving the heuristic's performance through the addition or deletion of variables makes logical sense. In fact, even when a new heuristic model is validated using the method just described, improvements to the model and it's classification or forecasting performance should still be sought via further modification of the input variable set.

Additional modification of an accepted new heuristic follows a path similar to that of comparing a new and existing heuristic. Additional variables may be added or removed to the accepted new heuristic model to develop an improved new heuristic model. This process may be repeated iteratively until all desired combinations of independent variables have been evaluated and the best performing heuristic model is kept (Swanson and White 1995, 1997). The addition or removal of variables in a step-wise fashion aides in determining the effect of individual variables or sets of variables on the heuristic decision model.

For example, a heuristic to evaluate the morbidity of a patient by utilizing four variables for heart rate, breath rate, perspiration, and skin temperature (Tahai et al. 1998) may be able to determine if the patient is in need of CPR (cardio-pulmonary resuscitation) with 93% accuracy, which appears to be a fairly good heuristic classification model. However, removing the last two variables: perspiration and temperature, increases the heuristic model's accuracy to almost 100%. Further reductions in the input variable vector set reduce the overall accuracy of the neural network model to approximately 50%. Addition of a systolic blood pressure variable to heart rate and breath rate variables would also serve to reduce the model's accuracy since systolic blood pressure is correlated with heart rate. Thus, even though an acceptable heuristic model is developed initially, further evaluation should still continue on the variable set used in the heuristic model to possibly improve performance.

Furthermore, neural network heuristic decision model refinement through the deletion of correlated or other unproductive variables assists in reducing the overall data collection costs associated with the neural network (Bansal et al. 1993). The reduction of data collection costs will be especially important in medical domains, if the remaining variables are available through less costly and especially less invasive medical test.

Next we will examine several applications of neural networks in evaluating heuristics and development of new heuristic methods.

10.4 Neural Network Evaluation of Medical Heuristics

Heuristics are frequently used in medical domains where rapid assessment and treatment reduces medical costs and may improve patient outcomes. Sophisticated tests exist that are capable of accurately gathering information about a patient's condition, but these tests are costly and take time to perform. The usage of neural networks to evaluate hypotheses (heuristics) of nonlinear components in exploratory medical data analysis has been previously suggested (Lisboa 2002). However, hypothesis testing with neural networks has thus far only been performed in an ad hoc manner without a formal methodology for evaluating new heuristics.

This section examines several previous research efforts that have been able to validate new decision heuristics or occasionally invalidate existing heuristics in medical resource allocation and diagnostic medical decision problems. The effective management of resources and patient information is critical for managing costs and improving the quality of patient care (Buchman et al. 1994). It should be noted that the original purpose for most of the reported research in this section was to create decision support tools for use in hospitals to manage resource allocation problems or improve diagnosis of specific medical and trauma conditions. The creation of new heuristic methods or the invalidation of existing heuristics occurred as a side effect of the original research, but serves to illustrate the usage of neural networks to evaluate new diagnostic or resource allocation decision heuristics.

Table 10.1 provides a summary of the neural network research that has indicated the development of a new heuristic or the invalidation of an existing heuristic method in a medical domain. The reported prior research in Table 10.1 utilized a variety of supervised learning algorithms and hidden layer architectures, thus demonstrating the robustness of supervised learning neural network training methods for heuristic decision model development and evaluation in medical domains. The training methods specifically include: one and two hidden layer backpropagation or multilayer perceptron, radial basis function, fuzzy ARTMAP, probabilistic neural networks, soft max discriminant analysis, self-organizing maps (an unsupervised learning method), and others.

The research listed in Table 10.1 are cases were a neural network model produced a new heuristic or modified the variable elements of an existing heuristic to create a better performing heuristic decision model. The research listed in Table 10.1 is not meant to be exhaustive (since well over 100 medical domain neural network journal research articles have been published in the last 2 years alone), but rather representative of much of the current medical domain research being done with neural networks.

Table 10.1. Samples of neural networks (NNs) discovering medical heuristics

Medical domain	Heuristic results	Citation
Blood/transfusion resource allocation	MSBOS heuristic previously used for determining blood needs has weaknesses, NNs are better at predicting required blood for transfusions	(Walczak and Scharf 2000)
	Information available upon arrival at an ER could be used to predict the transfusion needs of adult trauma patients	(Walczak 2005)
Brain/epilepsy/ head injury	NN using Doppler velocity variables accurately predicts head injuries	(Erol et al. 2005)
	Uses Lyapunov exponents with EEG to classify epilepsy	(Güler et al. 2005b)
	NNs used to show that EEG rhythmicity may be used to classify seizures as epileptic or nonepileptic	(Nowack et al. 2002; Walczak and Nowack 2001)
	NN analyzes very large amounts of time domain and frequency domain EEG data	(Srinivasan et al. 2005)
Breast cancer	NN utilizes new variables (urokinase-type plasminogen activator (uPA), and plasminogen activator inhibitor-type 1 (PAI-1)) combined with gene expression signatures for determining chemotherapy	(Harbeck et al. 2007)
Heart/heart disease/ circulation	NN uses a large number of available variables, but improves over physician and other IT methods	(Baxt 1991, 1995; Baxt and Skora 1996)
	Establishes a lower cutoff value for the troponin l variable	(Eggers et al. 2007)
	Uses signal/noise ratio to improve NN performance	(Güler and Übeyli 2005)
	Recommends inclusion of ECG data with standard perfusion scans for a more accurate model	(Gjertsson et al. 2006)
	NN used noninvasive variables only to diagnose heart disease	

Table 10.1. (continued)

Medical domain	Heuristic results	Citation
	Evaluated the use of genetic variant variables for predicting venous thrombosis	(Mobley et al. 2005) (Penco et al. 2005)
	Use of $FSPO_2$ and fetal heart rate variability variables with CTG tracings improves diagnosis of fetal hypoxia	(Salamalekis et al. 2006)
Injury severity/ length of stay	Using presentation data only, NN accurately predicted morbidity and length of stay for burn patients	(Frye et al. 1996)
	Using only data from the ER, NN predicts morbidity	(Izenberg et al. 1997)
	Accurate prediction of pediatric acuity of care with information available in the first 10 min of arrival at the ER	(Walczak and Scorpio 2000)
	NN uses noninvasive variables (spectrometry) to indicate burn severity	(Yeong et al. 2005)
Lung	NN uses of new technique, minimal-polyline approximation, to detect emphysema compared to curvature-based features, from chest radiographs	(Coppini et al. 2007)
	NN uses new technology to diagnose lung sounds	
	NN demonstrated that inclusion of a reactive glucose variable would improve the existing heuristic of using the d-dimer value in isolation to predict pulmonary embolism	(Güler et al. 2005a) (Walczak et al. 2006)
Pancreatitis (acute)	NN with less costly variables is able to outperform an abbreviated version of the RANSON score commonly used for predicting acute pancreatitis patient outcomes	(Pofahl et al. 1998; Walczak et al. 2003)
Prostate	NNs enable the use of lower PSA level to reduce false positives and reduce unnecessary biopsies	(Reckwitz et al. 1999)
	NN uses percent free prostate specific antigen variable to improve diagnosis of prostate cancer	(Stephan et al. 2002)

In order to examine the representative neural network research presented in Table 10.1, the cited research will be classified into three categories of new heuristic development:

- Development of a new heuristic when no competing heuristic exists
- Development of a new heuristic that utilizes significantly different variable elements from an existing heuristic
- Development of a new heuristic which is mostly an improvement of an existing heuristic, primarily through modification of the variable elements in the existing heuristic

Much of the research listed in Table 10.1 follows the evaluation methodology shown in Fig. 10.2 by creating the new heuristic model and determining it's performance through a neural network implementation. These neural network models are then compared statistically against the existing heuristic model to determine if the newer model improves overall performance. The neural network performance is evaluated against the previous models which may include statistical models, such as multiple regression, or expert physician performance on the same domain problem. Recall that in medical diagnostic domains, performance is typically compared through the comparison of sensitivity and specificity values or alternately by using another statistical comparison method such as receiver operator characteristic (ROC) curves or t-tests.

10.4.1 New Heuristic Development Without a Competing Heuristic

Development of a brand new heuristic decision model when another does not yet exist, may be seen as a special case of creating a new heuristic model that uses a significantly different set of variable elements for producing it's heuristic decisions. In this case, multiple neural network models are developed to determine the optimal variables for the heuristic and each model is compared against the others, with the currently best performing model serving as the existing heuristic model in Fig. 10.2.

Many of the earlier research projects listed in Table 10.1 fall into this class of a new heuristic model without any competing heuristic decision model (Baxt 1991, 1995; Baxt and Skora 1996; Frye et al. 1996; Izenberg et al. 1997). This is likely due to information processing for diagnostic purposes still being new to the medical field and heuristics to deal with the ever increasing quantity of patient information were only in early stages of development. These may also be viewed as proof of concept types of research that demonstrated a new tool, the neural network, for accurately evaluating large quantities of variables in a reliable manner, thus creating there own heuristic methodology.

Another motivation for development of brand new heuristic models is to satisfy the increasing pressure to reduce medical costs while maintaining or improving patient quality of care. The early prediction of trauma transfusion requirements NN utilizes information typically available at arrival in the emergency room (ER) without the need for invasive or costly laboratory tests (Walczak 2005). Mobley et al. (2005) demonstrate that a NN may accurately predict coronary disease using only variables available noninvasively. Neural networks have shown the ability to accurately diagnose the injury severity of pediatric trauma patients using only information available within the first 10 min of arrival in the ER (Walczak and Scorpio 2000). Finally, using a spectrometer and noninvasive data only, a neural network heuristic model has been able to accurately predict burn depth and severity (Yeong et al. 2005) This trend for developing diagnostic heuristics that utilize noninvasive test results or reduce the need for invasive procedure and thereby the risk to the patient is likely to continue as a driving goal in medical neural network research.

10.4.2 New Heuristic Development Competing Against an Existing Heuristic

The next classification of heuristic model development is for new heuristic models that compete against an existing heuristic model. Research for creating a new heuristic decision model that utilizes significantly different variables and also for modifying an existing decision heuristic to gain improved performance normally arises from dissatisfaction with the performance of the existing heuristic method or possibly simple intellectual curiosity to determine if any improvements are feasible.

Dissatisfaction with an existing heuristic, the MSBOS (Maximum Surgical Blood Order Supply) system, is the cause for the development of the transfusion prediction model that utilizes standard presurgery laboratory tests to significantly improve on blood unit ordering (Walczak and Scorpio 2000). As per Fig. 10.2, a neural network model utilizing physician determined variables was iteratively compared against the MSBOS across a large accumulation of historic transfusion data, with the neural network model consistently outperforming the old MSBOS heuristic when evaluated using a C/T ratio (units ordered/units transfused), by over 2 units average per patient. Refinements to the new heuristic were then accomplished using the leave-one-out technique described earlier to reduce the overall variable count for the neural network based heuristic transfusion resource model from 9 to 7 variables, which further decreased the C/T ratio by almost 0.7 units.

Another example, which actually disproves an existing heuristic method and recommends a new heuristic method came from a desire to see if neural networks were capable of improving predicting patient outcomes, in particular for acute pancreatitis (Pofahl et al. 1998; Walczak et al. 2003), over a more traditional method. In this neural network research to predict outcome severity with regard to hospital length of stay (LOS), numerous variables were identified and two sets of variables, with one being a subset of the first, were identified as predictive. Following the process in Fig. 10.2, each heuristic model was optimized and then compared against the other. The smaller variable set outperformed the larger set with evaluation being performed by comparing the mean absolute difference in LOS predicted by the neural network model vs. the actual LOS and percentage correct for 1–7, 8–14, and longer LOS. The larger variable set contained a commonly used heuristic variable, the abbreviated RANSON score, while the smaller set did not. Additional analysis was performed utilizing a regression model based on the abbreviated RANSON score as an alternate heuristic model. The abbreviated RANSON score was used, since a corollary focus of the research was on making the length of stay predictions utilizing information that was available within the first 12 h of presentation at an ER. The 22 variable LOS heuristic prediction model also outperformed the abbreviated RANSON-based regression model. This example raised questions about the current utilization of the RANSON scores as a valid heuristic for predicting acute pancreatitis patient outcomes.

The application of a new set of variables in medical decision heuristics may also result from advances in medical technology (Güler et al. 2005a), such as the inclusion of gene expression signatures for diagnosing breast cancer (Harbeck et al. 2007) and the combination of genetic variant variables to diagnose venous thrombosis (Penco et al. 2005). This is especially true with new imaging technologies, such as the utilization of Doppler velocity signals to accurately diagnose head trauma (Erol et al. 2005) and advances in blood imagining technology (Zini 2005). The research by Penco et al. (2005) also helps illustrate the new heuristic model refinement process of Fig. 10.2. After the new neural network heuristic model was convincingly shown to outperform existing heuristic models for diagnosing venous thrombosis, the new model was optimized by reducing the overall variable count from 62 values to 9 and finally down to just 3 variable values.

10.4.3 Improving an Existing Heuristic to Create a New Heuristic

The third heuristic development method seeks to improve the utilization and efficacy of existing heuristics instead of replacing them through the modification of the utilized variables in the heuristic model. This modification may be the addition of a new group of variables to a current set of variables or may involve the deletion of specific variables from an existing set of decision variables for a medical diagnostic or resource allocation heuristic. Deletion of highly correlated variables is necessary for neural networks models and will ultimately improve their classification or prediction performance.

A common theme in medicine, again, is reduction of costs and this may be achieved by reducing the unnecessary utilization of very expensive tests and treatments for patients that are falsely identified as belonging to a treatment group. Typically this is done by trying to improve the negative predictive capability of existing heuristics.

Several of the examples from Table 10.1 fall into this type of heuristic development, where improving the negative predictive capability of an existing heuristic is the goal of the research, with subsequent reduction in risk to patients and also a significant reduction in healthcare costs. Neural networks have been able to reduce false positives for: prostate cancer and subsequent reduction in unnecessary biopsies by being able to lower the PSA level at which a biopsy is ordered (Reckwitz et al. 1999), pulmonary embolism and reduction in chest radiography by adding in the reactive glucose variable with the existing d-dimer assay variable (Walczak et al. 2006), and earlier and more reliable detection of acute myocardial infarction by reducing the cutoff value for troponin l (Eggers et al. 2007). For each of these neural network heuristic diagnostic models incorporating the expanded variable set is compared, similar to Fig. 10.2, to either a separate neural network implementation or existing statistical model of the existing heuristic to demonstrate the efficacy of the new heuristic variable elements in improving diagnostic performance.

Another example of the heuristic refinement process from Fig. 10.2 for improving an existing heuristic is shown in the research by Güler and Übeyli (2005) who utilize a signal to noise ratio measurement to remove high noise variables from a neural network heuristic model to improve it's performance with a new smaller set of variable elements. As mentioned previously, the refinement of an existing heuristic through the removal of some variables further serves to reduce medical costs by reducing the data acquisition costs for the heuristic model (Bansal et al. 1993).

10.4.4 Methodological Heuristics

Each of the preceding medical domain examples has focused on the development of a better diagnostic or resource allocation heuristic method and involves modeling the new heuristic with a neural network implementation. A common claim made in the various research samples displayed in Table 10.1 is the efficacy and superiority of the neural network nonparametric modeling paradigm over more traditional statistical methods or domain expert performance. Much other neural network research is focused solely on demonstrating improvements to decision making through the utilization of neural network models. To distinguish these types of claims from the development of new heuristic models, this will be identified as improving the methodology or tool utilized to model the new problem solving heuristic, which may be seen as a fourth type of heuristic development where an existing heuristic model undergoes performance improvement without altering the variable elements of the heuristic model. These performance improvements are typically gained because the original model may have ignored some of the parametric requirements for the variables used in the heuristic method or a nonlinear interaction component between variable elements was present.

Academic research publications, especially in business, have failed to recognize the modeling power and statistical similarities of neural networks compared with other more commonly used parametric statistical models (e.g., regression and discriminant analysis). The equivalence of neural network modeling techniques compared to statistical modeling techniques has been shown in the literature for a wide variety of statistical techniques (Cheng and Titterington 1994; Raudys 1998; Zhang 2007), including: autoregression (Conner et al. 1994; Cottrell et al. 1995), canonical correlation analysis (Via et al. 2007), discriminant analysis (Gallinari et al. 1991), linear regression (Kumar 2005; Stern 1996; Warner and Misra 1996), logistic regression (Schumacher et al. 1996; Warner and Misra 1996), and maximum variance generalization (Via et al. 2007) among others.

As such, researchers frequently demonstrate the utility of their neural network models through comparison with traditional statistical models that utilize the same input variables. In Fig. 10.2 then, the existing heuristic would be a commonly used statistical model and the new heuristic would be the same model implemented as a neural network. It is important though to when comparing a neural network diagnostic or classification heuristic model against existing statistical models that an appropriate model is selected. Commonly used statistical models in medical domains include stepwise linear regression, logistic regression, and discriminant analysis (Walczak and Scorpio 2000; Zhang and Berardi 1998).

Neural networks have been shown to outperform both parametric and nonparametric statistical models across a wide variety of domains. Examples of superior neural network performance over traditional statistical methods are: in medical domains (Baxt and Skora 1996; Dybowski and Gant 1995; Lapuerta et al. 1995, León 1994; Razi and Athappilly 2005; Rodvold et al. 2001; Zhang and Berardi 1998) and business domains (Bansal et al. 1993; Devika and Achenie 1994; Falas et al. 1994; Lee et al. 1993; Piramuthu et al. 1994; Refenes 1993; Steurer 1993). These results were foretold by Patuwo et al. (1993) who demonstrate that using more sophisticated modeling techniques like neural networks can improve heuristic classification model performance by 15–22%.

The resulting heuristic-oriented research simply tries to improve upon the speed of the availability of diagnostic or resource information or improve performance without modifying the set of variables utilized by the current heuristic methods. Frequently this type of research may also argue for the utilization of neural network training methods other than the traditional multilayer perceptron backpropagation method. Examples of methodological-oriented heuristic improvement through translating an existing decision model into a neural network representation are: The use of cellular neural networks are proposed tom rapidly improve the speed of image processing for diagnosis of cancer (Arena et al. 2003), the use of probabilistic neural network models to improve performance in two separate medical diagnostic problems and also two protein localization problems (Georgiou et al. 2006), neural networks that automate the detection of metastases in bone scans with very high sensitivity and specificity (Sadik et al. 2006), a neural network model produced an 8% improvement in sensitivity for predicting ischemic heart disease (Scott et al. 2004), accuracy for predicting carotid artery stenosis was improved through use of a neuron-fuzzy system (Übeyli and Güler 2005a), and a mixture of experts (multiple) neural network system was able to obtain an almost 99% accuracy performance for diagnosing breast cancer (Übeyli and Güler 2005b). In addition to a neural network application providing the desired improvements, this type of research may also produce recommendations for combining neural network heuristic models with other applications, such as the combination of a neural network with a decision tree model to more accurately predict cancer relapse (Jerez-Aragones et al. 2003) and a combination of a neural network to refine EEG tracings with an expert system (Castellaro et al. 2002).

10.5 Neural Network Heuristic Evaluation in Business Domains

Businesses outside of the medical domain also face complex decision making problems where heuristic techniques are applied to simplify or speed up the decision making process. The process for evaluating heuristics in more general business domains is identical to the heuristic evaluation method previously described and shown in medical domains.

This section will start by listing business-oriented neural network research that specifically claims to create new heuristics or improve upon existing heuristics, similar to how neural network heuristic models were shown in the medical heuristics section. Table 10.2 lists neural network heuristic research in business domains.

Because of the frequent application of neural networks in solving business and engineering decision problems, some interesting aspects of utilizing neural networks for heuristic development emerge in the business domains. Both statistics and neural networks have been the primary methods for evaluating credit card applications and risks (He et al. 2004) and as such the neural network models may serve as the existing model in Fig. 10.2 for comparison to new methods including new neural network techniques. Besides serving as the existing heuristic model for new research, neural networks have also been proposed as an efficient methodology for selecting between existing and competing heuristics (Gupta et al. 2000), which would mean that the neural network is serving as a heuristic method to select the most appropriate decision heuristic, which could potentially include other neural network heuristic models.

A few select neural network heuristic development cases from Table 10.2 are now analyzed in greater detail to demonstrate the neural network heuristic comparison methodology shown in Fig. 10.2. The first two examples represents the evaluation using neural networks of multiple existing heuristics that compete within a domain. The competing heuristics are: utilization of technical vs. fundamental analysis models, the financial heuristic that “more

Table 10.2. Samples of neural network (NN) heuristic research in business

Business domain	Heuristic development type[a]	Heuristic result	Citation
Accounting	3 (or 4)	Establishes bridge to economic accelerator models	(Fioretti 2004)
	2	NN used to develop new methodology for creating bankruptcy prediction models recommending representative samples vs. stratified samples for model training	(Wilson and Sharda 1994)
Finance	4	Uses PCA component analysis to improve NN predictions	(Ince and Trafalis 2007)
	4	NN demonstrates futility of day trading stable stocks	(Tsang et al. 2007)
	3	NN validates global knowledge needed for predicting emerging market financial time series	(Walczak 1999, 2004)
	2	NN shows minimal knowledge needed to predict financial time series	(Walczak 2001)
	3	NN reduces loan default for lender by 10%	(West 2000)
Human Resources	2	NN improves scheduling of airline ground staff	(Hao et al. 2004)
	1	NN utilized to predict enrollment patterns to schedule admissions personnel	(Walczak 1998; Walczak and Sincich 1999)
Management	2	NN used to optimize load allocations for cogeneration power plants	(Cerri et al. 2006)
	3	NN optimizes nuclear fuel management	(Jiang et al. 2006)
Manufacturing	4 (or 3)	NN combines greedy and nongreedy heuristics to improve job scheduling	(Agarwal et al. 2006)
	4	NN improves flow shop scheduling using an existing heuristic	(Haq and Ramanan 2006)
	4	Combination of NN and decision trees reduce false classification of control chart patterns	(Ruey-Shiang 2005)

Table 10.2. (continued)

Business domain	Heuristic development type[a]	Heuristic result	Citation
	2	NN improves on real-time job shop scheduling	(Shugang et al. 2005)
	2	NN minimizes exceptional elements through fractional cell formation in group technology	(Venkumar and Haq 2006)
	2	NN combines flow shop and parallel machine algorithms to produce new heuristic for flexible flow shop scheduling	(Wang et al. 2003)
Marketing	3	Determining heuristic variables with independent component analysis produces NN that outperforms all other statistical methods	(Ahn et al. 2007)
	1	NN uses navigation patterns to accurately predict customer product knowledge, if combined with more traditional surveys, even greater improvement is possible	(Chang et al. 2006)
	4	NN significantly improves service discontinuation predictions over multiple discriminant analysis models	(Walczak and Parthasarathy 2006)
Transportation	2	NN improves the responsiveness and reliability of paratransit systems	(Fu and Teply 1999)
	4	NN improves traffic predictions over other heuristic methods	(Ishak and Alecsandru 2004)
	4	NN as a part of an Expert System maximizes air freight shipping	(Lau et al. 2004)

[a] 1 = New heuristic w/o a competing heuristic, 2 = new heuristic w/competing heuristic, 3 = modification of existing heuristic, 4 = methodological improvement of existing heuristic

information produces better models", and the development of bankruptcy prediction models using representative or stratified data sets.

Neural network models, with the addition of some variables, that represent both technical and fundamental models are implemented and evaluated against each other. The various fundamental and technical models implemented in neural networks serve as the two heuristic models for comparison in Fig. 10.2. Since the two models are competing on an equal basis, with neither one considered.

The existing model, refinement of the independent variables for each model is performed simultaneously and reevaluation of all new models is performed. The model selection research methodology (Swanson and White 1995, 1997) is used to select the best performing neural network heuristic model, with the evaluation for both competing models being performed across the same data set for the dependent variables.

Neural network financial time series, especially in foreign exchange rate predictions, typically utilize a technical and homogeneous model, utilizing a single lag[1] equal to the time period being forecast (e.g., one day lags for a single day forecast or a five-day lag for a one week forecast). Neural network research has shown that using a more fundamental analysis type of model, which includes values for macroeconomic variables outside of the exchange rate itself would improve forecasting capabilities of neural network heuristic exchange rate prediction models over a more technical model (Walczak 2001). A corollary outcome of this research showed that financial time series are cyclical and that training forecasting models with data beyond the first full cycle was unnecessary, which contradicted the existing financial ideal of "the more data the better".

This heuristic evaluation result of fundamental or heterogeneous models outperforming technical or homogeneous models was further exploited in research that demonstrated how global financial indicators are required of heuristic models attempting to forecast stock market index futures in emerging markets in the Pacific Rim (Walczak 1999) and South America (Walczak 2004). In each case, both technical and fundamental analysis heuristic models were developed and evaluated against each other, with the fundamental heuristic models consistently demonstrating superior forecasting performance and the forecasting performance exceeding traditional random walk models of financial forecasting. The fund

However, technical models still tend to dominate neural network and information systems models of financial markets in general. This is because the models themselves are simpler due to the reliance on only a single variable or homogenous set of variables derived from the original forecasting value, which reduces variable costs (Bansal et al. 1993) and ultimately makes the

[1] A lag is the difference between a financial value at one time, typically the day prior to the forecast, and another past day. A one day lag for a financial value v at time t is $v_t - v_{t-1}$ and a five day lag for the same value at time t is $v_t - v_{t-5}$.

heuristic model's results easier to interpret. Other research has focused on the second heuristic development method described in Sect. 4.4, the improvement of an existing heuristic through the addition or deletion of variables, to maximize the performance of these technical models. The reported research (Walczak 2001; Walczak et al. 1998) demonstrates that utilizing multiple lag values significantly improves the forecasting ability of these neural network heuristic models. Again, it should be emphasized that the evaluation of these heuristics is done by statistically analyzing the forecasting performance of the single day lag neural network model against other neural network models utilizing multiple lag values in the input vector set and also statistical models (e.g., ARIMA).

The final example concerns the propensity for bankruptcy models to be constructed utilizing stratified data sets (typically a 50–50 distribution). Early research indicated that heuristic bankruptcy prediction neural network models could be optimized using representative training data, meaning that the training data reflected the group proportionality of the real world (Hu et al. 1996; Wilson and Sharda 1994). These neural network models evaluating the possibility of representative training samples were evaluated against similar neural network models using various stratified training samples and also against statistical models (e.g., logistic regression) also with representative and stratified samples. All results for evaluation were performed against a representative data sample to reflect utilization of these heuristic models This type of finding would make data collection for model building easier since it could be done automatically without performing any matching to stratify the training (model development) data sets. However, other research has more recently contradicted these findings, indicating that neural networks and statistical models need to be trained on a 50/50 stratified sample to optimize performance by keeping the neural network from becoming trapped in a local minimum (Sharda and Wilson 1996). Again, the more recent research duplicated the methodology of simultaneously comparing multiple neural network and statistical models developed using both representative and stratified training samples, only with a different data set.

Like the medical heuristic development through neural networks, the examples described in this section and given in Table 10.2 demonstrate that neural networks are a reliable and effective mechanism for evaluating and developing business related heuristics. However, as the last case has pointed out, car must be taken when generalizing results and new heuristics should be validated across multiple data sets before ultimately replacing an existing heuristic.

10.6 Conclusions

Neural networks are commonly used in business (Smith and Gupta 2000; Wong et al. 2000) and medicine (Baxt 1995; Dybowski and Gant 1995; Zini 2005). Typically, new neural network applications implement heuristic methods for new problems or to improve performance over existing decision heuristics. Neural networks, because of their nonparametric nature serve as an outstanding evaluation tool for comparison of heuristic models and evaluation of new heuristic models. Lewin (2000) goes so far as to state that neural networks are a cross-over methodology between artificial intelligence and regression.

This chapter has defined a more formal process for utilizing neural networks in heuristic decision model evaluation. Several examples of how the process has been applied in previous research in both medical domains and business domains have been presented to demonstrate the application of the described process. The process is meant to serve as a guide to researchers and professional developers for evaluating competing heuristic decision models utilizing neural networks. One or both of the competing models may be implemented as a neural network, but comparison against traditional nonneural network heuristic and statistical models is applicable.

Feng et al. (2006) examine the development and evaluation of a regression and neural network model in the manufacturing business domain. There process of developing two competing models, based on existing heuristic methods, and then evaluating them statistically to compare the results and select the optimal model is similar to the proposed methodology formalism in this chapter. Ultimately in their research, the neural network and regression models were shown to be equivalent and the new heuristic model (which is actually what this chapter calls a methodological heuristic advancement) implementation selection is based on other factors, which in this case is the belief that neural networks are a black-box methodology.

Researchers must realize that neural network models are as rigorous as more traditional statistical models and have distinct advantages for modeling complex and potentially nonlinear business and medical decision making heuristics, such as tolerance of noise in the data, fast training and real-time results, and nonparametric capability to model numerous population and error distributions.

References

Abdel-Aal, R.E. (2004) Experimental evaluation of performance improvements in abductive network classifiers with problem decomposition. *Neurocomputing* 61(1–4), 193–215

Agarwal, A., Jacob, V.S., and Pirkul, H. (2006) An improved augmented neural-network approach for scheduling problems. *INFORMS Journal on Computing* 18(1), 119–128

Ahn, H., Choi, E., and Han, I. (2007) Extracting underlying meaningful features and canceling noise using independent component analysis for direct marketing. *Expert Systems with Applications* 33(1), 181–191

Arena, P., Basile, A., Bucolo, M., and Fortuna, L. (2003) Image processing for medical diagnosis using CNN. *Nuclear Instruments and Methods in Physics Research Section A* 497(1), 174–178

Ballini, R. and Gomide, F. (2002/2003) Heuristic learning in recurrent neural fuzzy networks. *Journal of Intelligent and Fuzzy Systems* 13(2–4), 63–74

Bansal, A., Kauffman, R.J., and Weitz, R.R. (1993) Comparing the modeling performance of regression and neural networks as data quality varies: A business value approach. *Journal of Management Information Systems* 10(1), 11–32

Baxt, W.G. (1991) Use of an artificial neural network for the diagnosis of myocardial infarction. *Annals of Internal Medicine* 115, 843–848

Baxt, W.G. (1995) Application of artificial neural networks to clinical medicine. *The Lancet* 346(8983), 1135–1138

Baxt, W.G. and Skora, J. (1996) Prospective validation of artificial neural network trained to identify acute myocardial infarction. *The Lancet* 347, 12–15

Buchman, T.G., Kubos, K.L., Seidler, A.J., and Siegforth, M.J. (1994) A comparison of statistical and connectionist models for the prediction of chronicity in a surgical intensive care unit. *Critical Care Medicine* 22(5), 750–762

Castellaro, C., Favaro, G., Castellaro, A., Casagrande, A., Castellaro, S., Puthenparampil, D.V., and Fattorello Salimbeni, C. (2002) An artificial intelligence approach to classify and analyse EEG traces. *Clinical Neurophysiology* 32(3), 193–212

Cerri, G., Borghetti, S., and Salvini, C. (2006) New management for heat and power cogeneration plants. *Engineering Applications of Artificial Intelligence* 19(7), 721–730

Chang, S.E., Changchien, S.W., and Huang, R.-H. (2006) Assessing users' product-specific knowledge for personalization in electronic commerce. *Expert Systems with Applications* 30(4), 682–693

Cheng, B. and Titterington, D.M. (1994) Neural networks: A review from a statistical perspective. *Statistical Science* 9(1), 2–30

Connor, J.T., Martin, R.D., and Atlas, L.E. (1994) Recurrent neural networks and robust time series prediction. *IEEE Transactions on Neural Networks* 51(2), 240–254

Coppini, G., Miniati, M., Paterni, M., Monti, S., and Ferdeghini, E.M. (2007) Computer-aided diagnosis of emphysema in COPD patients: Neural-network-based analysis of lung shape in digital chest radiographs. *Medical Engineering and Physics* 29(1), 76–86

Cottrell, M., Girard, B., Girard, Y., Mangeas, M., and Muller, C. (1995) Neural modeling for time series: A statistical stepwise method for weight elimination. *IEEE Transactions on Neural Networks* 6(6), 1355–1364

Daniel, W.W. (1999) *Biostatistics: A Foundation for Analysis in the Health Sciences*. New York: Wiley

Devika, P.D. and Achenie, L.E.K. (1994) Experience with Using a Neural Network for Forecasting Time Series. In: *World Congress on Neural Networks – San Diego Volume 2*, San Diego, CA, pp. 315–320

Dybowski, R. and Gant, V. (1995) Artificial neural networks in pathology and medical laboratories. *The Lancet* 346, 1203–1207

Efron, B. (1982) *The Jackknife, the Bootstrap and Other Resampling Plans*. Philadelphia, PA: Siam

Eggers, K.M., Ellenius, J., Dellborg, M., Groth, T., Oldgren, J., Swahn, E., and Lindahl, B. (2007) Artificial neural network algorithms for early diagnosis of acute myocardial infarction and prediction of infarct size in chest pain patients. *International Journal of Cardiology* 114(3), 366–374

Erol, F.S., Uysal, H., Ergün, U., Barişçi, N., Serhathoğlu, S., and Hardalaç, F. (2005) Prediction of minor head injured patients using logistic regression and MLP neural network. *Journal of Medical Systems* 29(3), 205–215

Falas, T., Charitou, A., and Charalambous, C. (1994) The Application of Artificial Neural Networks in the Prediction of Earnings. In: *Proceedings of IEEE International Conference on Neural Networks Volume VI*, Orlando, FL, pp. 3629–3633

Feng, C.-X.J., Yu, Z.-G., Kusiak, A. (2006) Selection and validation of predictive regression and neural network models based on designed experiments. *IIE Transactions* 38(1), 13–23

Fioretti, G. (2004) The investment acceleration principle revisited by means of a neural network. *Neural Computing and Applications* 13(1), 16–23

Frye, K.E., Izenberg, S.D., Williams, M.D., and Luterman, A. (1996) Simulated biologic intelligence used to predict length of stay and survival of burns. *Jounral of Burn Care and Rehabilitation* 17(6), 540–546

Fu, L. and Teply, S. (1999) On-line and off-line routing and scheduling of dial-a-ride paratransit vehicles. *Computer-Aided Civil and Infrastructure Engineering* 14(5), 309–319

Gallinari, P., Thiria, S., Badran, R., and Fogelman-Soulie, F. (1991) On the relationships between discriminant analysis and multilayer perceptrons. *Neural Networks* 4(3), 349–360

Georgiou, V.L., Pavlidis, N.G., Parsopoulos, K.E., Alevizos, P.D., and Vrahatis, M.N. (2006) New self-adaptive probabilistic neural networks in bioinformatic and medical tasks. *International Journal of Artificial Intelligence Tools* 15(3), 371–396

Gjertsson, P., Lomsky, M., Richter, J., Ohlsson, M., Tout, D., van Aswegen, A., Underwood, R., and Edenbrandt, L. (2006) The added value of ECG-gating for the diagnosis of myocardial infarction using myocardial perfusion scintigraphy and artificial neural networks. *Clinical Physiology and Functional Imaging* 26(5), 301–304

Güler, İ., Polat, H., and Ergün, U. (2005a) Combining neural network and genetic algorithm for prediction of lung sounds. *Journal of Medical Systems* 29(3), 217–231

Güler, İ. and Übeyli, E.D. (2005) Feature saliency using signal-to-noise ratios in automated diagnostic systems developed for ECG beats. *Expert Systems with Applications* 28(2), 295–304

Güler, N.F., Übeyli, E.D., and Güler, İ. (2005b) Recurrent neural networks employing Lyapunov exponents for EEG signals classification. *Expert Systems with Applications* 29(3), 506–514

Gupta, J.N.D., Sexton, R.S., and Tunc, E.A. (2000) Selecting scheduling heuristics using neural networks. *INFORMS Journal on Computing* 12(2), 150–162

Hao, G., Lai, K.K., and Tan, M. (2004) A neural network application in personnel scheduling. *Annals of Operations Research* 128(1–4), 65–90

Haq, A.N. and Ramanan, T.R. (2006) A bicriterian flow shop scheduling using artificial neural network. *International Journal of Advanced Manufacturing Technology* 30(11/12), 1132–1138

Harbeck, N., Schmitt, M., Paepke, S., Allgayer, H., and Kates, R.E. (2007) Tumor-associated proteolytic factors uPA and PAI-1: Critical appraisal of their clinical relevance in breast cancer and their integration into decision-support algorithms. *Critical Reviews in Clinical Laboratory Sciences* 44(2), 179–201

He, J., Liu, X., Shi, Y., Xu, W., and Yan, N. (2004) Classification of credit cardholder behavior by using Fuzzy linear programming. *International Journal of Information Technology and Decision Making* 3(4), 633–650

Hertz, J., Krogh, A., and Palmer, R.G. (1991) *Introduction to the Theory of Neural Computation*. Reading, MA: Addison-Wesley

Hu, M.Y., Hung, M.S., Shanker, M.S., and Chen, H. (1996) Using neural networks to predict performance of Sino-foreign joint ventures. *International Journal of Computational Intelligence and Organizations* 1(3), 134–143

Ince, H. and Trafalis, T.B. (2007) Kernel principal component analysis and support vector machines fro stock price prediction. *IIE Transactions* 39(6), 629–637

Ishak, S. and Alecsandru, C. (2004) Optimizing traffic prediction performance of neural networks under various topological, input, and traffic condition settings. *Journal of Transportation Engineering* 130(4), 452–465

Izenberg, S.D., Williams, M.D., and Luterman, A. (1997) Prediction of trama mortality using a neural network. *American Surgeon* 63(3), 275–281

Jain, A.K., Mao, J., and Mohiuddin, K.M. (1996) Artificial neural networks: A tutorial. *IEEE Computer* 29(3), 31–44

Jerez-Aragones, J.M., Gomez-Ruiz, J.A., Ramos-Jimenez, G., Munoz-Perez, J., and Albaconejo, E. (2003) A combined neural network and decision trees model for prognosis of breast cancer relapse. *Artificial Intelligence in Medicine* 27(1), 45–63

Jiang, S., Ziver, A.K., Carter, J.N., Pain, C.C., Goddard, A.J.H., Franklin, S., and Phillips, H.J. (2006) Estimation of distribution algorithms for nuclear reactor fuel management optimization. *Annals of Nuclear Energy* 33(11/12), 1039–1057

Kumar, U.A. (2005) Comparison of neural networks and regression analysis: A new insight. *Expert Systems with Applications* 29(2), 424–430

Lapuerta, P., Azen, S.P., and LaBree, L. (1995) Use of neural networks in predicting the risk of coronary artery disease. *Computers in Biomedical Research* 28(1), 38–52

Lau, H.C.W., Choy, K.L., Lau, P.K.H., Tsui, W.T., and Choy, L.C. (2004) An intelligent logistics support system for enhancing the airfreight forwarding business. *Expert Systems* 21(5), 253–268

Lee, T., White, H., and Granger, C. (1993) Testing for neglected nonlinearity in time series models. *Journal of Econometrics* 56(3), 269–290

León, M.A. (1994) Binary response forecasting: comparison between neural networks and logistic regression analysis, In: *World Congress on Neural Networks*, San Diego, CA, pp. 244–247

Lewin, D.I. (2000) Getting clinical about neural networks. *IEEE Intelligent Systems and Their Applications* 15(1), 2–5

Lim, T., Loh, W., and Yu-Shan, S. (2000) A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning* 40(3), 203–228

Lisboa, P.J.G. (2002) A review of evidence of health benefit from artificial neural networks in medical intervention. *Neural Networks* 15(1), 11–39

Lisboa, P.J. and Taktak, A.F.G. (2006) The use of artificial neural networks in decision support in cancer: A systematic review. *Neural Networks* 19(4), 408–415

Markey, M.K., Tourassi, G.D., Margolis, M., and DeLong, D.M. (2006) Impact of missing data in evaluating artificial neural networks trained on complete data. *Computers in Biology and Medicine* 36(5), 516–525

McLachlan, G.J. (1992) *Discriminant Analysis and Statistical Pattern Recognition*. New York: Wiley

Mobley, B.A., Schechter, E., Moore, W.E., McKee, P.A., and Eichner, J.E. (2005) Neural network predictions of significant coronary artery stenosis in men. *Artificial Intelligence in Medicine* 34(2), 151–161

Nath, R., Rajagopalan, B., and Ryker, R. (1997) Determining the saliency of input variables in neural network classifiers. *Journal of Computers and Operations Research* 24(8), 767–773

Nowack, W.J., Walczak, S., and Janati, A. (2002) Clinical correlate of EEG rhythmicity. *Journal of Clinical Neurophysiology* 19(1), 32–36

Pakath, R. and Zaveri, J. S. (1995) Specifying Critical Inputs in a Genetic Algorithm-driven Decision Support System: An Automated Facility. *Decision Sciences* 26(6), 749–779

Patuwo, E., Hu, M.Y., and Hung, M.S. (1993) Two-group classification using neural networks. *Decision Sciences* 24(4), 825–845

Pearl, J. (1984) *Heuristics – Intelligent Search Strategies for Computer Problem Solving*. Reading, MA: Addison-Wesley

Penco, S., Grossi, E., Cheng, S., Intraligi, M., Maurelli, G., Patrosso, M.C., Marocchi, A., and Buscema, M. (2005) Assessment of the Role of Genetic Polymorphism in Venous Thrombosis Through Artificial Neural Networks. *Annals of Human Genetics* 69(6), 693–706

Piramuthu, S., Shaw, M., and Gentry, J. (1994) A classification approach using multi-layered neural networks. *Decision Support Systems* 11(5), 509–525

Pofahl, W.P., Walczak, S.M., Rhone, E., and Izenberg, S.D. (1998) Use of an artificial neural network to predict length of stay in acute pancreatitis. *American Surgeon* 64(9), 868–872

Raudys, S. (1998) Evolution and generalization of a single neuron – I: Singlelayer perceptron as seven statistical classifiers. *Neural Networks* 11(2), 283–296

Razi, M.A. and Athappilly, K. (2005) A comparative predictive analysis of neural networks (NNs), nonlinear regression and classification and regression tree (CART) models. *Expert Systems with Applications* 29(1), 65–74

Reckwitz, T., Potter, S.R., Snow, P.B., Zhang, Z., Veltri, R.W., and Partin, A.W. (1999) Artificial neural networks in urology: Update 2000. *Prostate Cancer and Prostatic Diseases* 2(5/6), 222–226

Refenes, A.N. (1993) Constructive Learning and Its Application to Currency Exchange Rate Forecasting. In: R. Trippi and E. Turban (Eds.), *Neural Networks in Finance and Investing*, New York: Irwin, pp. 465–493

Rodvold, D.M., McLeod, D.G., Brandt, J.M., Snow, P.B., and Murphy, G.P. (2001) Introduction to artificial neural networks for physicians: Taking the lid off the black box. *The Prostate* 46(1), 39–44

Ruey-Shiang, G. (2005) Real-time pattern recognition in statistical process control: a hybrid neural network/decision tree-based approach. *Proceedings of the Institution of Mechanical Engineers – Part B – Engineering Manufacture* 219(3), 283–298

Sadik, M., Jakobsson, D., Olofsson, F., Ohlsson, M., Suurkula, M., and Edenbrandt, L. (2006) A new computer based decision support system for the interpretation of bone scans. *Nuclear Medicine Communications* 27(5), 417–423

Salamalekis, E., Siristatidis, C., Vasios, G., Saloum, J., Giannaris, D., Chrelias, C., Prentza, A., and Koutsouris, D. (2006) Fetal pulse oximetry and wavelet analysis of the fetal heart rate in the evaluation of abnormal cardiotocography tracings. *Journal of Obstetrics and Gynaecology Research* 32(2), 135–139

Schumacher, M., Robner, R., and Vach, W. (1996) Neural networks and logistic regression – Part I. *Computational Statistics and Data Analysis* 21(6), 661–682

Schürmann, J. (1996) *Pattern Classification a Unified View of Statistical and Neural Approaches*. New York: Wiley

Scott, J.A., Aziz, K., Yasuda, T., and Gewirtz, H. (2004) Integration of clinical and imaging data to predict the presence of coronary artery disease with the use of neural networks. *Coronary Artery Disease* 15(7), 427–434

Sharda, R. and Wilson, R.L. (1996) Neural network experiments in business-failure forecasting: Predictive performance measurement issues. *International Journal of Computational Intelligence and Organizations* 1(2), 107–117

Shin, K.S., Lee, T.S., and Kim, H. (2005) An application of support vector machines in bankruptcy prediction model. *Expert Systems with Applications* 28(1), 127–135

Shugang, L., Zhiming, W., and Xiaohong, P. (2005) Job shop scheduling in real-time cases. *International Journal of Advanced Manufacturing Technology* 26(7/8), 870–875

Smith, M. (1993) *Neural Networks for Statistical Modeling*, New York: Van Nostrand Reinhold

Smith, K.A. and Gupta, J.N.D. (2000) Neural networks in business: techniques and applications for the operations researcher. *Computers and Operations Research* 27(11–12), 1023–1044

Srinivasan, V., Eswaran, C., and Sriraam, N. (2005) Artificial Neural Network Based Epileptic Detection Using Time-Domain and Frequency-Domain Features. *Journal of Medical Systems* 29(6), 647–660

Soulié, F. (1994) Integrating Neural Networks for Real World Applications. In: Zurada, Marks, and Robinson (Eds.), *Computational Intelligence: Imitating Life*, New York: IEEE Press, pp. 396–405

Stephan, C., Jung, K., Cammann, H., Vogel, B., Brux, B., Kristiansen, G., Rudolph, B., Hauptmann, S., Lein, M., Schnorr, D., Sinha, P., and Loening, S.A. (2002) An artificial neural network considerably improves the diagnostic power of percent free prostate-specific antigen in prostate cancer diagnosis: results of a 5-year investigation. *International Journal of Cancer* 99(3), 466–473

Stern, H.L. (1996) Neural networks in applied statistics. *Technometrics* 38(3), 205–214

Steurer, E. (1993) Nonlinear Modelling of the DEM/USD Exchange Rate. In: A. Refenes (Ed.), *Neural Networks in the Capital Markets*, New York: Wiley, pp. 199–211

Swanson, N.R. and White, H. (1995) A model-selection approach to assessing the information in the term structure using linear models and artificial neural networks. *Journal of Business and Economic Statistics* 13(3), 265–275

Swanson N.R. and White, H. (1997) A model selection approach to real-time macroeconomic forecasting using linear models and artificial neural networks. *Review of Economics and Statistics* 79(4), 540–550

Tahai, A., Walczak, S., and Rigsby, J.T. (1998) Improving Artificial Neural Network Performance Through Input Variable Selection. In: P. Siegel, K. Omer, A. deKorvin, and A. Zebda (Eds.) *Applications of Fuzzy Sets and the Theory of Evidence to Accounting II*, Stamford, Connecticut: JAI Press, pp. 277–292

Tourassi, G.D. and Floyd, C.E. (1997) The effect of data sampling on the performance evaluation of artificial neural networks in medical diagnosis. *Medical Decision Making* 17(2), 186–192

Tsang, P.M., Kwok, P., Choy, S.O., Kwan, R., Ng, S.C., Mak, J., Tsang, J., and Koong, K. (2007) Design and implementation of NN5 for Hong Kong stock price forecasting. *Engineering Applications of Artificial Intelligence* 20(4), 453–461

Übeyli, E.D. and Güler, I. (2005a) Adaptive neuro-fuzzy inference systems for analysis of internal carotid arterial Doppler signals. *Computers in Biology and Medicine* 35(8), 687–702

Übeyli, E.D. and Güler, I. (2005b) A mixture of experts network structure for breast cancer diagnosis. *Journal of Medical Systems* 29(5), 569–579

Venkumar, P. and Haq, A.N. (2006) Fractional cell formation in group technology using modified ART1 neural networks. *International Journal of Advanced Manufacturing Technology* 30(11/12), 1132–1138

Via, J., Santamaría, I., and Pérez, J. (2007) A learning algorithm for adaptive canonical correlation analysis of several data sets. *Neural Networks* 20(1), 139–152

Walczak, S. (1998) Neural network models for a resource allocation problem. *IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics* 28(2), 276–284

Walczak (1999) Gaining competitive advantage for trading in emerging capital markets with neural networks. *Journal of Management Information Systems* 16(2), 177–192

Walczak, S. (2001) An empirical analysis of data requirements for financial forecasting with neural networks. *Journal of Management Information Systems* 17(4), 203–222

Walczak, S. (2004) Forecasting Emerging Market Indexes with Neural Networks. In: G.P. Zhang (Ed.), *Neural Networks in Business Forecasting*, Hershey, PA: Idea Group Publishing, pp. 80–101

Walczak, S. (2005) Artificial neural network medical decision support tool: Predicting Transfusion Requirements of ER Patients. *IEEE Transactions on Information Technology in Biomedicine* 9(3), 468–474

Walczak, S., Brimhall, B.B., and Lefkowitz, J.B. (2006) Nonparametric decision support systems in medical diagnosis: Modeling pulmonary embolism. *International Journal of Healthcare Information Systems and Informatics* 1(2), 65–82

Walczak, S. and Nowack, W.J. (2001) An artificial neural network approach to diagnosing epilepsy using lateralized bursts of theta EEGs. *Journal of Medical Systems* 25(1), 9–20

Walczak, S. and Parthasarathy, M. (2006) Modeling online service discontinuation with nonparametric agents. *Information Systems and e-Business Management* 4(1), 49–70

Walczak, S., Pofahl, W.E., and Scorpio, R.J. (2003) A decision support tool for allocating hospital bed resources and determining required acuity of care. *Decision Support Systems* 34(4), 445–456

Walczak, S. and Scharf, J.E. (2000) Reducing surgical patient costs through use of an artificial neural network to predict transfusion requirements. *Decision Support Systems* 30(2), 125–138

Walczak, S. and Scorpio, R.J. (2000) Predicting pediatric length of stay and acuity of care in the first ten minutes with artificial neural networks. *Pediatric Critical Care Medicine* 1(1), 42–47

Walczak, S. and Sincich, T. (1999) A comparative analysis of regression and neural networks for university admissions. *Information Sciences* 119(1–2), 1–20

Walczak, S., Tahai, A., and Khondar, K. (1998) Improved Cash Flows Using Neural Network Models for Forecasting Foreign Exchange Rates. In: P. Siegel, K. Omer, A. deKorvin, and A. Zebda (Eds.) *Applications of Fuzzy Sets and The Theory of Evidence to Accounting II*, Stamford, Connecticut: JAI Press, pp. 293–310

Wang, H., Jacob, V., and Rolland, E. (2003) Design of efficient hybrid neural networks fro flexible flow shop scheduling. *Expert Systems* 20(4), 208–231

Warner, B. and Misra, M. (1996) Understanding neural networks as statistical tools. *The American Statistician* 50(4), 284–293

Weigand, A.S. and Zimmermann, H.G. (1995) The Observer-Observation Dilemma in Neuro-Forecasting: Reliable Models From Unreliable Data Through CLEARNING. *Proceedings of Artificial Intelligence Applications on Wall Street*, New York, pp. 308–317

West, D. (2000) Neural Network Credit Scoring Models. *Computers and Operations Research* 27(11/12), 1131–1152

Wilson, R.L. and Sharda, R. (1994) Bankruptcy prediction using neural networks. *Decision Support Systems* 11(5), 545–557

Wong, B.K., Lai, V.S., and Lam, J. (2000) A bibliography of neural network business applications research: 1994–1998. *Computers and Operations Research* 27(11–12), 1045–1076

Yeong, E.-K., Hsiao, T.-C., Chiang, H.K., and Lin, C.-W. (2005) Prediction of burn healing time using artificial neural networks and reflectance spectrometer. *Burns* 31(4), 415–420

Zahedi, F. (1996) A meta-analysis of financial applications of neural networks. *International Journal of Computational Intelligence and Organizations* 1(3), 164–178

Zhang, G.P. (2007) Avoiding pitfalls in neural network research. *IEEE Transactions on Systems, Man, and Cybernetics – Part C* 37(1), 3–16

Zhang, G.P. and Berardi, V.L. (1998) An investigation of neural networks in thyroid function diagnosis. *Health Care Management Science* 1(1), 29–37

Zhang, G.P. and Hu, M.Y. (1998) Neural network forecasting of the British pound/US dollar exchange rate. *Omega, International Journal of Management Science* 26(4), 495–506

Zini, G. (2005) Artificial intelligence in Hematology. *Hematology* 10(5), 393–400

11

Building Intelligent Sensor Networks with Multiagent Graphical Models

Yang Xiang

Computing and Information Science University of Guelph, Canada
yxiang@cis.uoguelph.ca

Summary. Modern society relies heavily on various equipments. To ensure productive operation, avoid downtime and reduce maintenance cost, engineers must constantly determine whether equipment is operating normally and what is the small set of devices that is highly likely the culprit of abnormality. A sensor network is often deployed to gather and process the key information in this decision process. With the traditional centralized approach for sensor network monitoring, the data transmission can introduce delay, the centralized processing can create a bottleneck, and the central unit must have access to all the knowledge needed. The intelligent sensor network is a promising alternative, where a set of distributed agents embody local sensors, local computing resources, local knowledge, and inference procedures, and cooperate through limited communication. This chapter introduces, at the application development level, the approach based on multiply sectioned Bayesian networks, a probabilistic framework for agent inference in intelligent sensor networks. Through a case study, key technological steps involved in applying the framework are linked together and practitioners are facilitated in mapping theoretical intricacies to practical reality.

11.1 Introduction

Modern society relies heavily on various equipments (food production processes, assembly lines, transportation vehicles, airplanes, electricity grids, etc.) Consider monitoring a piece of complex equipment. To ensure productive operation, avoid downtime and reduce maintenance cost, engineers must constantly determine whether the equipment is operating normally. If the equipment is determined to be abnormal, the faulty devices must be replaced. Very often, a detected abnormal behavior of the equipment may be caused by one or more faulty devices from a large number of candidates. It is simply too costly to replace them all. Hence, the next decision is to determine a small number of devices that, if replaced, is highly probable to bring the equipment back to normal. This is equivalent to answer the query: what is the small set of devices that is highly likely the culprit of abnormality?

Y. Xiang: *Building Intelligent Sensor Networks with Multiagent Graphical Models*, Studies in Computational Intelligence (SCI) **97**, 289–320 (2008)
www.springerlink.com

A sensor network is often deployed to gather and process the key information in this decision process. Complex equipment consists of multiple components, each of which is further composed of multiple devices. For some devices, sensors can be deployed to observe their inputs and outputs, but sensor observations are noisy and unreliable. For other devices, no sensors can be deployed to observe their inputs and outputs due to accessibility or cost. No sensor can directly observe whether a device is normal and hence the state of the device must be inferred. We refer to sensors deployed in the equipment and their transmission media collectively as the *sensor network*. To infer the state of a device from sensor observations, knowledge about intended and faulty behavior of the device as well as knowledge about other devices it interfaces with are necessary. Often, components are manufactured by different vendors, who may be unwilling to disclose internals of their components. In such cases, no single entity has all the knowledge needed: an issue that arises when the sensor network crosses technical and economical boundaries.

Traditional approach for sensor network monitoring is centralized, where all sensor observations are transmitted to a central location for processing. Transmission introduces delay; centralized processing creates a bottleneck; and the central unit must have access to all the knowledge needed.

As the cost of computing and networking continues to decrease, distributed processing becomes a more promising alternative, where sensor observations are processed locally and processing units exchange only partial information through message passing. Each processing unit is abstracted as an intelligent *agent*, embodying its subset of sensors, its computing resources, its local knowledge, and its inference procedures. The collection of these agents as well as the sensors that they manage forms an *intelligent sensor network*. The task to process sensor observations by these agents and to answer the query "what is the small set of highly probable faulty devices" becomes a task of *multiagent inference*.

Alternative approaches exist for multiagent inference. The early approach is logic-based. Logic has intrinsic limitations in handling uncertainty. Effort to overcome these limitations leads to default logic based approach. This approach relies on default and model minimization to handle uncertainty. There are situations, however, where the minimal model is not the most probable. More recent approach is based on Markov decision processes (MDPs) for its strength in handling uncertainty. It suffers, however, from high computational complexity.

In this chapter, we introduce the approach based on multiply sectioned Bayesian networks (MSBNs). Built upon the success of Bayesian Networks (BN) (Pearl 1988; Neapolitan 1990; Shafer 1996; Castillo et al. 1997; Cowell et al. 1999; Jensen 2001), MSBNs (Xiang 2002) provide a probabilistic framework for reasoning about uncertain domains in cooperative multiagent systems (MAS). Under the framework, a complex, uncertain problem domain is partitioned into overlapping subdomains so that each can be managed by a single intelligent agent. The agent holds a partial perspective of the domain in terms of a Bayesian subnet over the subdomain. These agents reason autonomously as well as through limited communication. The distributed inference operations defined under the MSBN framework ensure that their beliefs are *exact* as governed by Bayesian probability theory. These beliefs form a distributed assessment of the current state of the domain and answer the query "'what is the small set of highly probable faulty devices" in the context of sensor network. When the subnet dependency structures are sparse, the inference computation is *efficient*.

Several advances have been made in recent years on modeling, compilation and inference under the MSBN framework, making it even closer to field applications. Before a general technological framework can be turned into deployed applications, practitioners must understand sufficiently well how theoretical intricacies are mapped into practical reality. The levels of such understanding can be described as follows:

1. Mathematical and algorithmic level.
2. Application development level.
3. Operation level.

This chapter is intended to facilitate understanding at the application development level. It links together key technological steps involved in applying the MSBN framework to intelligent sensor networks through a case study (in a laboratory setting).

The problem domain of case study is a moderately sized sensor network for monitoring a combinational digital system. The choice of a digital system is due to the common knowledge (among readers) on digital circuits. Once how to apply the MSBN framework to such a system is understood, its applications to monitoring other equipment (electrical, mechanical, chemical or other nature) will be within one's grasp. We demonstrate how such a problem domain can be modeled as MSBN-based MAS, how the model can be compiled into an efficient run-time representation, and how agents can cooperate to monitor the digital system and isolate faults. We explain intuitively the rationales behind each operation. The operations are demonstrated using a state of the art software toolkit, WebWeavr, developed by the author and freely available to researchers and educators.

To serve its purpose, the chapter is kept as informal as possible, with pointers to references containing mathematical and algorithmic details. In short, the chapter addresses the following questions: What technical steps are involved in building an MSBN-based intelligent sensor network? Why are these steps necessary? How can these steps be performed using a software toolkit, such as WebWeavr? How does the resultant intelligent sensor network answer the query "what is the small set of highly probable faulty devices"? What are the benefits of adopting this framework?

The remainder of the chapter is organized as follows: Section 11.2 surveys the literature. Section 11.3 specifies the problem domain of the case study. Section 11.4 describes the knowledge representation and integration of the

MSBN-based MAS for the case study. Section 11.5 presents MAS system verification. Enhancement of agent interface for improved inference efficiency is addressed in Sect. 11.6. How to compile the MAS into an efficient run-time representation is demonstrated in Sect. 11.7. The decision making computation, how multiagent inference isolates faulty devices, is illustrated in Sects. 8 and 9.

11.2 Related Work

Several alternative frameworks for generic multiagent inference exist. The earliest one is the *blackboard* architecture (Nii 1986), a distributed rule-based system. It is essentially a logic-based approach.

The BDI architecture (Rao and Georgeff 1991) has been very influential in building MAS. It primarily deals with representation of an agent's mental state for practical reasoning where an agent's belief is represented as atoms of first-order logic (Wooldridge 2002).

The main limitations of logic in handling uncertainty are summarized by Russell and Norvig (2003) as the following: (1) Logic relies on exhaustive disjunction to ensure exceptionless rules and such disjunction can become unbounded in practical uncertain domains. (2) In practical uncertain domains, existing knowledge is not deterministic as suitably represented by logic rules. (3) In practical decision making, it is often too costly to gather all the facts needed for firing the necessary logic rules.

The limitations of logic lead to several extensions known as *default logic* (Reiter, 1980), *circumscription* (McCarthy 1980), *nonmonotonic logic* (McDermott and Doyle 1980), and *truth maintenance systems* (Doyle 1979). Distributed assumption-based truth maintenance system (DATMS) (Mason and Johnson 1989) and distributed truth maintenance system (DTMS) (Huhns and Bridgeland 1991) extend centralized truth maintenance systems to distributed agents. The basic idea is to begin the knowledge base with a set of default assumptions which are uncertain. Inference proceeds as if these assumptions were true until some are found to be false due to new observations. The falsified assumptions as well as conclusions drawn from them will be retracted from the knowledge base to regain consistency. The computational complexity of these extensions is at least NP-hard (Russell and Norvig 2003).

Some frameworks have been developed specially for data fusion in sensor networks. Roos et al. (2003) propose a multiagent framework for sensor network monitoring based on logical consistency. They showed that establishing a global diagnosis under the framework is NP-Hard and therefore their protocol does not guarantee one.

Guestrin et al. (2004) propose distributed regression for efficient interpretation of sensor data. Their method assumes that the data can be fit into a linear function.

None of the above frameworks maintains agents' beliefs in terms of Bayesian probability. A recent trend has focused on modeling multiagent decision making using MDP (Boutilier 1999; Xuan et al. 2000; Nair 2005). However, it has been shown (Bernstein et al. 2000) that the computation for solving general distributed MDPs is intractable. The state of the art algorithms can handle only very small problem domains currently (much smaller than the domain in the case study to be presented here).

In the remainder of this chapter, we present a case study based on the MSBN framework. The key advantages of the framework are the following: It maintains agents' beliefs in terms of exact Bayesian probability and is well suited to modeling of uncertain domains. It works well with nonlinear dependence relations among sensor observations. It guarantees globally consistent diagnosis. Due to its graphical modeling, the computation is efficient when the graphical structure is sparse.

11.3 Sensor Network for Digital System Monitoring

Our case study involves monitoring a combinational digital system. It consists of remotely located components $U_0, \ldots, U_4$ supplied by five independent venders and integrated by a sixth vendor. Each component is further composed of a number of devices (logic gates) as shown in Figs. 11.1–11.5.

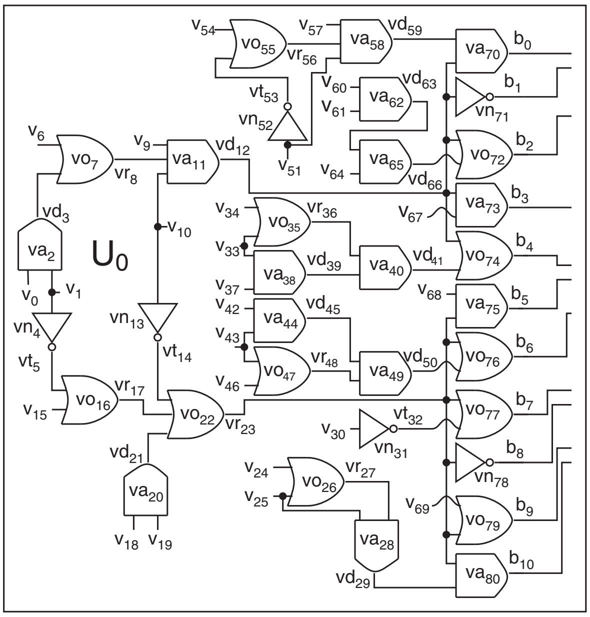


Fig. 11.1. Component U_0

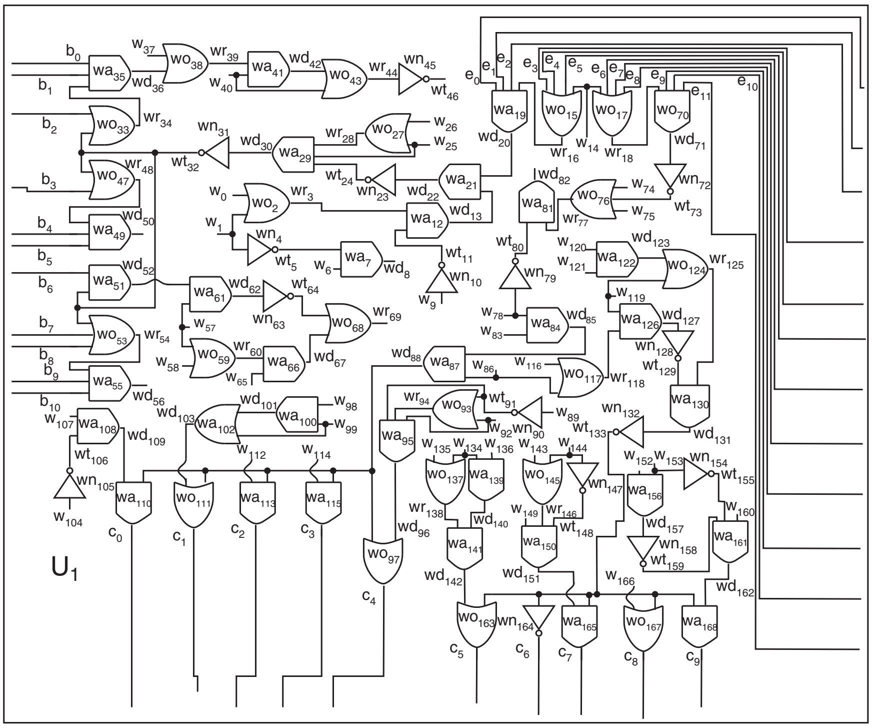


Fig. 11.2. Component U_1

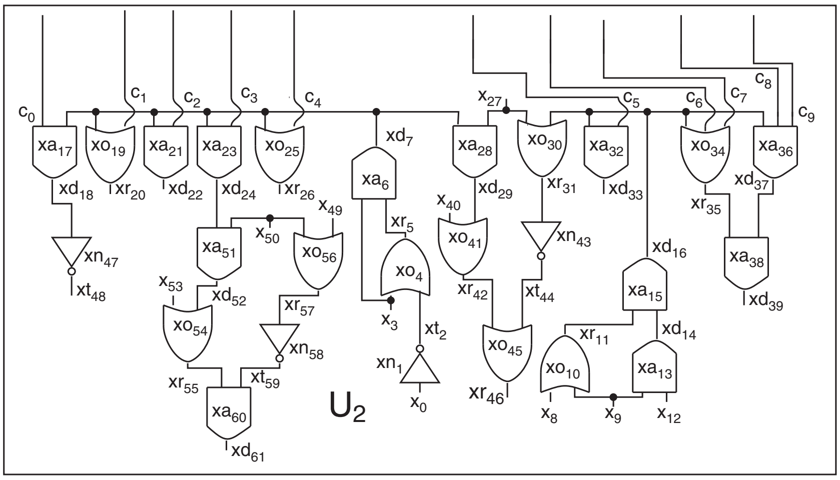


Fig. 11.3. Component U_2

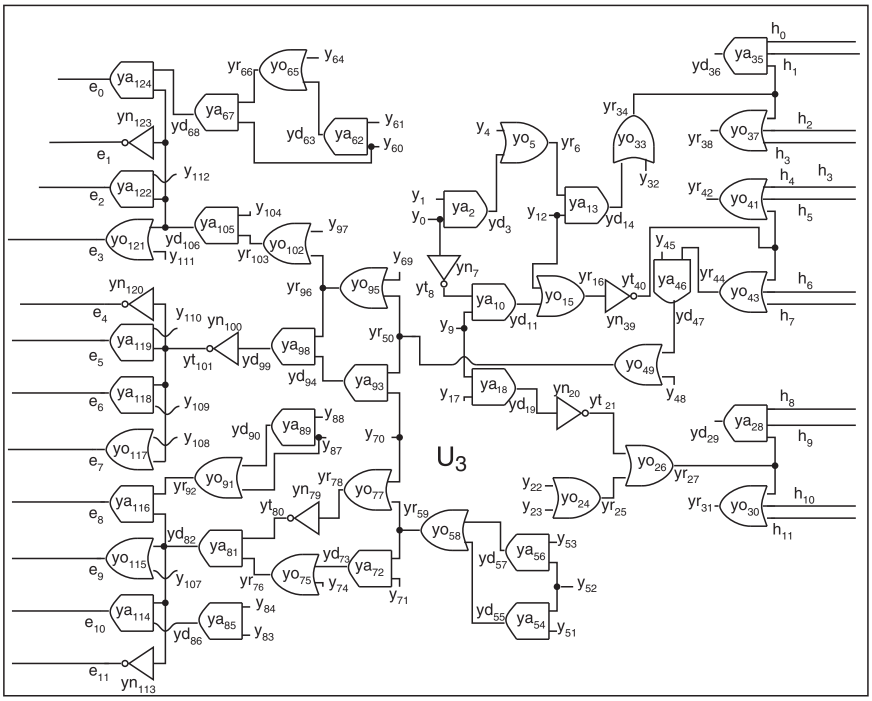


Fig. 11.4. Component U_3

Each component has some external inputs, such as signal v_{54} (see top of Fig. 11.1) in U_0. It may accept signals from other components. For instance, U_1 accepts signal b_0 (see top left of Fig. 11.2) from U_0 (see top right of Fig. 11.1). It may output signals to others. For example, U_1 outputs signal c_0 (see bottom left of Fig. 11.2) to U_2 (see top left of Fig. 11.3).

Signals exchanged between components are labeled identically, e.g., $c_0, \ldots, c_9$ between U_1 (see bottom of Fig. 11.2) and U_2 (see top of Fig. 11.3). In the case study, all signals are assumed binary (taking values of logic 0 or 1). In general, each signal can take a finite number of discrete values, or even be continuous (see, for instance, Moral et al. (2001)).

Each device is in one of two states, normal or faulty, although more states can also be represented (e.g., two normal states each at a different operating mood). We assume that each device may be in the faulty state at any given time with a probability of 0.01. A faulty NOT gate produces incorrect output 50% of time. The corresponding probabilities for AND and OR gates are assumed to be 0.8 and 0.3, respectively. These parameters and Figs. 11.1–11.5 *completely* specify the problem domain and allow replication/verification of our case study. These details also give readers a feel of the complexity of the problem domains that the MSBN framework is capable of handling. The digital system is used here as an example of any complex system made of

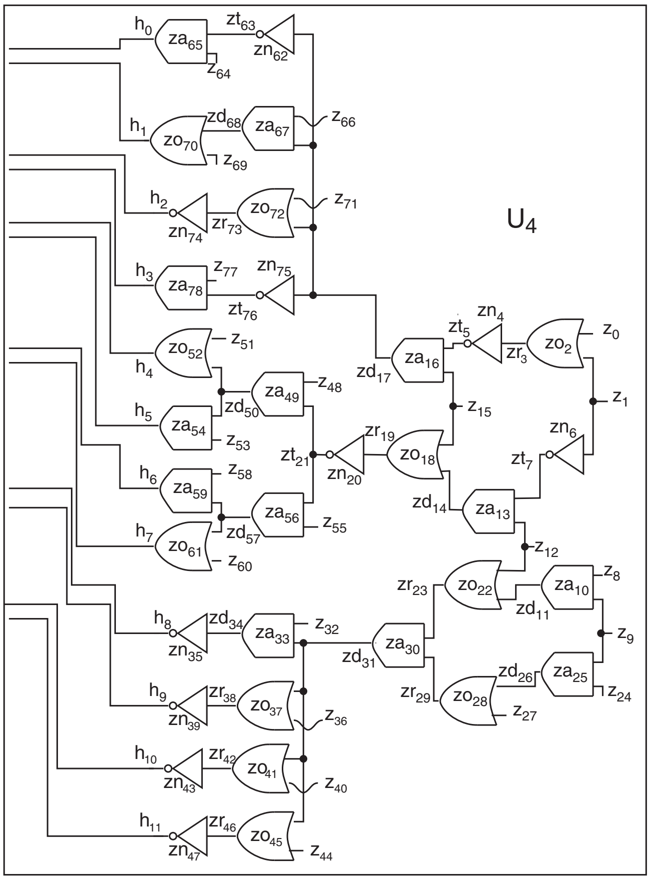


Fig. 11.5. Component U_4

multiple components, each of which further decomposes into simpler units. Collectively, these components implement some useful functionality that may be electrical, mechanical, chemical, and so on.

There are two types of decisions to be made in monitoring such a system. The first decision involves a *normality* query: Is the current system operating normally? The answer to the query is binary. A positive answer (the system is normal) requires no intervention. A negative answer (the system is abnormal) requires intervention in order to bring the system back to normal.

In the case of a negative answer, the second decision must be made to decide which faulty gates should be replaced. Very often, a faulty gate produces incorrect output signal which propagates to other gates and causes their output signals to be incorrect as well. It is simply too costly and unwise to replace them all. It is desirable to replace only a small number of devices that are highly probable to be the culprit of abnormality. Hence, the second decision involves a *culprit* query: What is the small set of devices that is highly likely the culprit of abnormality? The answer to this query has multiple potential values.

To answer these queries, sensors can be deployed to collect necessary raw information. We assume that in general each external input signal and the output signal of each gate can be observed through a sensor, although in practice no sensors are deployed to observe some signals due to cost involved or other constraints. We assume that whether a logical gate is faulty can never be observed directly and can only be inferred from other observations. These assumptions are consistent with the partial observability of practical systems. We refer to the collection of sensors and the signal transmission media deployed to monitor a given system as a *sensor network*.

Although a sensor network provides the raw information about the behavior of the monitored system, the information must be processed in order to answer the normality query and culprit query. In theory, the processing can be performed by an intelligent agent through reasoning based on its knowledge on the monitored system as well as sensor observations. However, this paradigm has a number of practical difficulties. First of all, transmitting sensor observations distributed over space to a central location requires high bandwidth and introduces time delay. Second, the dependence on a single agent creates a bottleneck and processing fails completely when the agent fails. Third, the need to process all relevant knowledge and observations at the single agent places heavy load of computation at the agent and introduces additional computational delay. Fourth, it is difficult to develop a single agent capable of monitoring a large and complex problem domain, due to the amount of domain knowledge to be gathered and encoded. Very often, no single person or single technical entity possesses all the knowledge needed. For example, we have assumed that the digital system consists of five components supplied by independent vendors. Although each vendor has detailed knowledge about the composition of the corresponding component, it may not want to disclose this knowledge due to competition.

One alternative to the single-agent paradigm is the multiagent paradigm. The large domain is partitioned into subdomains. In our case study, each subdomain corresponds to one component. Sensor observations for each subdomain are collected and processed by a separate agent, and a set of agents are responsible for the entire domain. The advantages can be understood as follows: First, the agent responsible for a given component can be deployed at the same location as the component, eliminating the need of high bandwidth and time delay due to transmission of observations to a central location. Second, even when a single agent fails, the other agents can still function. Hence, the monitoring system fails gracefully instead completely. Third, each agent needs to process mainly knowledge and observations on its subdomain (plus some communication with other agents). Since different agents process their local information in parallel, the overall computation is more efficient. Fourth, each agent encodes only knowledge about its subdomain and the agent development is easier. When the subdomains are partitioned naturally, a natural technical entity exists to supply the relevant knowledge. For our case study, the vendor who supplies the component becomes the natural agent developer.

In this chapter, we adopt the multiagent paradigm for the case study. The agent responsible for the component U_i ($i = 0, \ldots, 4$) is denoted by A_i. We refer to the collection of the sensor network, the local signal transmission media, and the agents as an *intelligent sensor network*.

11.4 Integration of MSBN-Based Multiagent System

Given the knowledge on a problem domain and sensor observations, agents can answer the normality and culprit queries based on their beliefs. For example, if every agent believes that each gate in its subdomain is currently normal, then agents collectively can answer the normality query positively. On the other hand, if at least one agent believes that some gates in its subdomain is currently abnormal, then agents collectively can answer the normality query negatively. In that case, the set of gates, each of which is believed highly probable to be faulty by at least one agent, constitutes the answer to the culprit query.

How, then, should agents represent their beliefs? It has been shown (Cox 1946) that under some reasonable assumptions, the correct belief must be consistent with Bayesian probability. Furthermore, if one's belief deviates from Bayesian probability, then actions consistent with that belief will lead to guaranteed failure in an malicious uncertain environment (de Finetti 1937). Belief maintained by earlier multiagent reasoning systems does not satisfy this criterion (see Sect. 11.2). The MSBN framework is developed with the objective that agents' beliefs are *exact* according to Bayesian probability theory. To this end, the framework employs two levels of knowledge representation: the individual agent level and the agent society level.

At the individual agent level, the knowledge is represented as a BN as in the single-agent paradigm. Under that paradigm, a BN is a concise encoding of the single agent's probabilistic knowledge of its domain through a graphical model. Under the multiagent paradigm, since each agent only has the knowledge about a subdomain and encodes that knowledge into a BN, the graphical model is referred to as the *subnet* of the agent.

A subnet consists of three components: a set of variables, a graph, and a set of conditional probability distributions (CPT). The set of variables corresponds to the *subdomain* of the agent. The graph is a directed acyclic graph (DAG), where each node corresponds to a subdomain variable (hence we refer to the nodes and variables interchangeably) and each arc corresponds to a causal dependence relation. The DAG encodes conditional independence relations among the variables. If two sets X and Y of nodes are graphically separated by a third set Z, then the dependency between X and Y is mediated by Z. Once the value of Z is known, X is no longer dependent on Y, and X

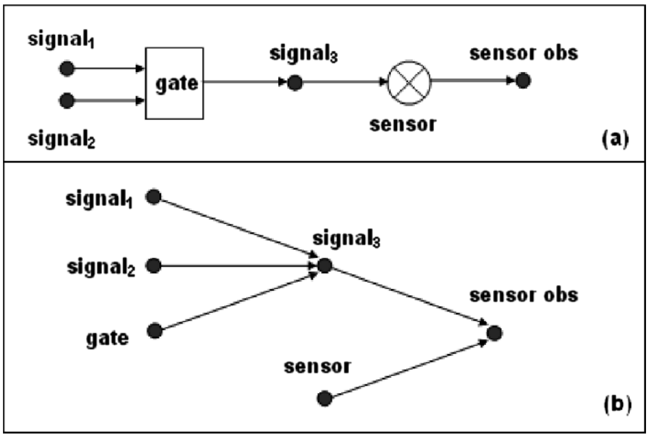


Fig. 11.6. (**a**) Illustration of a logic gate and a sensor. (**b**) Representation in a subnet

and Y are said to be *conditionally independent* given Z. The set of probability distributions consists of one CPT for each node x in the form of $P(x|\pi(x))$, where $\pi(x)$ is the parent nodes of x. Due to the encoding of conditional independence in the DAG, a probability distribution over all subdomain variables is well defined as the product of all CPTs. For fundamentals on representation of conditional independence in BNs, see (Pearl 1988; Neapolitan 1990; Shafer 1996; Castillo et al. 1997; Cowell et al. 1999; Jensen 2001).

In general, the subdomain of an agent in our case study may contain the following types of variables: *gate, signal, sensor* and *sensor observation*. Figure 11.6a illustrates a logic gate, its input and output signals, the sensor that monitors the gate output, and the sensor observation. How to encode the dependency among these variables in a subnet is shown in (b).

The illustration and the representation capture the unreliability of the sensor: When the sensor is functioning normally, the sensor observation is identical to the gate output ($signal_3$). When the sensor fails, its output may differ from that of the gate. From this, we see that sensors and logic gates that they monitor are not much different. They are both devices that are subject to failure and can be modeled in the same way.

One might point out their difference in observability: The output signal of the gate cannot be directly perceived by the agent, but the sensor observation can. However, although the sensor observations are directly perceivable, the perception requires the sensor output to be transmitted to the agent, which takes time and bandwidth. When a large number of sensors exist, the agent must choose the sensors to perceive selectively. Those sensors not being chosen at a given time are effectively not observable. We now see that sensors and devices (logic gates) that they monitor are not any different at all, from the modeling perspective. To simply our presentation, we assume the following in the case study:

- *All sensors are reliable.*
  Hence, sensor observations are always equal to the signals they monitor. This makes representation of sensors and sensor observations redundant. We therefore omit the sensor and sensor observation types of variables from the subnet and regard the corresponding signal variables directly observable subject to the following restriction.
- *Some signals do not have associated sensors.*
  This assumption divides signal variables into those that are observable and those that are not. Due to the omission of sensor variables, the difference

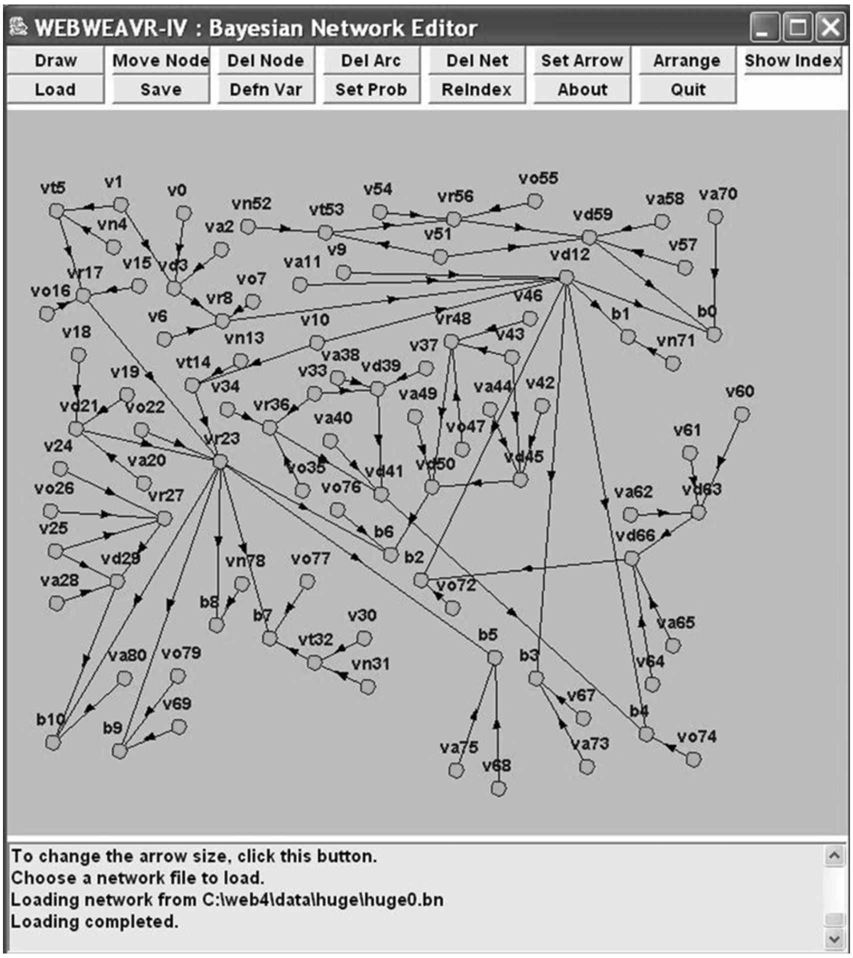


Fig. 11.7. Subnet S_0 for component U_0

vt5 Done

0	1	vn4	v1
0.5	0.5	bad	0
0.5	0.5	bad	1
0.0	1.0	good	0
1.0	0.0	good	1

Fig. 11.8. CPT for node vt_5 in subnet S_0

is not explicit in the subnet representation. We make observability as the default and we indicate explicitly when a signal variable is not observable.

Figure 11.7 shows the subnet S_0 (for agent A_0) constructed by the vendor of U_0 using the tool Network Editor from the WebWeavr software toolkit. It shares common nodes b_0 through b_{10} with subnet S_1 (not shown) for agent A_1. Due to the above assumption, the subdomain V_0 of S_0 consists of only *gate* and *signal* variables. A gate variable represents the state of a digital gate: whether it is normal or faulty (simply denoted as *good* and *bad*). For instance, vn_4 (see middle left of Fig. 11.1) represents a NOT gate. A signal variable represents the logic value of a signal if it is not observed by a sensor or the correctly observed value of the signal by a sensor. In either case, its value is either logic 0 or logic 1. For instance, vt_5 represents the sensed output signal of gate vn_4. The knowledge encoded in S_0 is private to its developer, the vendor of U_0. For simplicity, we say that S_0 is private to agent A_0 and this privacy will be maintained through the lifetime of A_0, as we will see. The exception is the variables that A_0 shares with other agents, e.g., b_0 through b_{10}. Since these variables are known to another component and its vendor, they are public anyway.

Figure 11.8 illustrates the CPT for variable vt_5. The last two rows, where vn_4 = good, encode the knowledge on the normal behavior of NOT gate vn_4. The two rows where vn_4 = bad state that, when the NOT gate is faulty, its output is random.

Next, we consider the knowledge representation at the agent society level. The key issues addressed at this level are agent organization and agent interface. Agent organization specifies, for each agent, which other agents it can communicate directly. Agent interface specifies, for each pair of agents who can communicate directly, what are the pubic variables between them as these variable determine the content of messages they exchange. Agent organization is presented in the remainder of this section. Agent interface is partly described here and is continued into Sects. 5 and 6.

The chief concerns of agent organization are to support exact and efficient probabilistic inference. Through formal analysis, it has been shown that the organization must be a underdirected tree structure (Xiang and Lesser 2003) (called a *hypertree*). In the hypertree, each hypernode corresponds to an agent and each hyperlink corresponds to a direct communication link between the agents connected (through their interface). That is, according to the organization, each agent can only communicate directly with agents adjacent on the hypertree.

Intuitively, in a hypertree organization, each hyperlink defines two separate agent communities (one on each end) and potentially allows information to be fully exchanged between the two communities through exactly two messages over the hyperlink (one in each direction). For a society of n agents, this amounts to exactly $2(n-1)$ messages along the hypertree, which is efficient. If agent interfaces are adequately composed (as will be addressed in Sect. 11.5), such message passing can also ensure exact probabilistic inference.

Who is responsible to specify the agent organization? As we mentioned, there exists a sixth independent vendor, referred to as *Assembler*, who assembles the five components into the final digital system. Assembler is also the natural candidate to assemble the five corresponding agents into an MAS. Operationally, it uses the tool Integrator from the WebWeavr toolkit illustrated in Fig. 11.9 (through the function buttons "Agt Org" and "Name Agt" at the top left of the figure). In the figure, the hypertree topology is shown where each hypernode is labeled with the nickname of an agent (*huge0* for A_0, *huge1* for A_1, and so on).

Next, we consider the agent interface. The chief concerns here are to support exact and efficient probabilistic inference, and to protect agent privacy. The efficiency and privacy concerns demand that a minimum amount of information to be exchanged between agents. The exactness concern demands that a sufficient amount of information to be exchanged over an interface. Recall that an agent interface (corresponding to a hyperlink in the hypertree) is the unique information channel between the two agent communities. Mathemat-

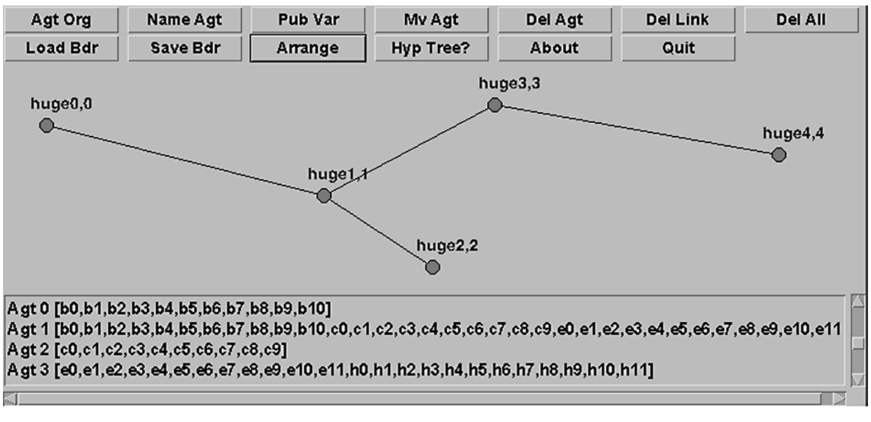


Fig. 11.9. Integrating agents into an MAS. *Middle*: agent organization. *Bottom*: agent public variables

ical analysis (Xiang and Lesser 2003) shows that the agent interface should consist of a set of variables that renders the two agent communities conditionally independent, and a message from an agent should be the agent's subjective probability distribution over the interface variables. This message contains all relevant information to inform the other agent community. Anything less is not sufficient in general.

The agent interfaces are specified through the tool Integrator by specifying, for each agent, a set of *public* variables (using the "Pub Var" function button at top of Fig. 11.9). From the set of public variables in each agent, the agent interfaces can be derived as the intersection of these variables between adjacent agents. For instance, A_0 has public variables $b_0, \ldots,$ b_{10}: signals exchanged between U_0 and U_1 (see the first line in the bottom of Fig. 11.9).

From the general requirement of the agent interface, a number of implied requirements can be derived and are enforced by the tool Integrator. Each public variable in an agent must be associated with at least another adjacent agent (otherwise, the variable is not really public). For each pair of adjacent agents, the two corresponding sets of public variables must have a nonempty intersection (otherwise, the content of message to be exchanged between them is undefined). If nonadjacent agents A_i and A_j have a common public variable, then it must be a public variable in each agent along the hypertree pathway between A_i and A_j. Otherwise, information from A_i on the variable cannot be communicated to A_j. This is because any information to be delivered between them must be conveyed indirectly through the agents between them on the hypertree, as dictated by the agent organization. The tool Integrator automatically enforces the above requirements during specification and gives feedbacks to Assembler until all conditions mentioned above are satisfied.

The MAS is now *logically* specified. However, in order for agents to communicate according to the specified organization, the MAS must be set up *physically*. That is, for each agent to be able to communicate directly to its hypertree neighbors, it must know their physical addresses in the computer network. To do so, Assembler uses the tool Binder from WebWeavr:

Binder is a special agent and its physical address is known to all agents. When it starts, it is given access to the organization specification of the MAS. It then waits for each agent to register. An agent registers itself by sending its physical address to Binder. For instance, agent A_0 (nicknamed *huge0*) sends its host computer IP address and port number to Binder. After all agents have registered, Binder notifies each of them with the physical address of each adjacent agent on the hypertree as well as the set of public variables shared between them, the agent interface.

The successful termination of the binding and agent registration mark the integration of the MSBN-based MAS. Each agent now knows to whom it can communicate directly, how to reach them, and what message content should be exchanged with them.

11.5 Model Verification

To ensure exact inference, the knowledge representation of the MAS must satisfy two additional conditions. Both conditions are related to the directions of arcs in agents' subnets. One of them has a global scope and the other is restricted to interface variables only.

First, we consider the global condition. When agents' subnets are viewed as a whole (by merging their public variables), it must be a DAG. This requirement is implied by the causal interpretation of the graphical structure of subnets. If we start from a variable in a subnet, traverse subnets through a directed path, and finally return to the same variable, then the graphical structure has violated the causal interpretation.

As we mentioned before, each subnet is a DAG, specified by the corresponding agent developer. However, when multiple DAGs (one for each subnet) are merged together, it may be cyclic. This is illustrated in Fig. 11.10 {cycle}. When the three DAGs G_1, G_2 and G_3 are merged through their public nodes (labeled identically in each DAG), a directed path (a, c, d, b, n, k, g, j, l, a) is formed.

The possible cyclicity from merging multiple DAGs means that just ensuring acyclicity at each subnet is not sufficient. That is, the global acyclicity cannot be enforced at the level of individual agent developers. Verification performed at the level of agent society is necessary. However, as each subnet is *private* (built by an independent vendor), the global acyclicity cannot be verified by physically merging individual subnets (which requires disclosure of the internal structure of each subnet to a centralizing agent). Despite its seeming impossibility, a verification method has been developed (Xiang 1998) that only requires each agent to pass messages to their hypertree neighbors on whether its subnet contains any parent or child of their shared variables (but not how many and what they are). Nothing else about the internal structure of its subnet is disclosed. Based on such messages, agents can cooperate to detect global cyclicity whenever it occurs and to verify global acyclicity whenever it holds. The verification tool DVerify in WebWeavr toolkit implements the method, whose operation will be briefly illustrated below.

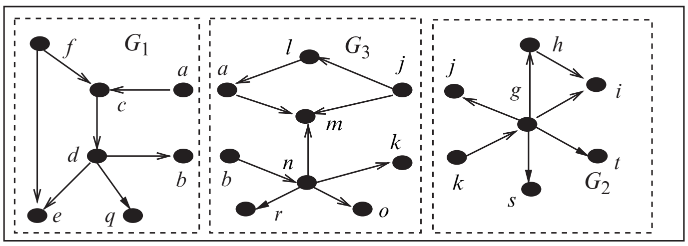


Fig. 11.10. A directed cycle is formed after the three DAGs are merged

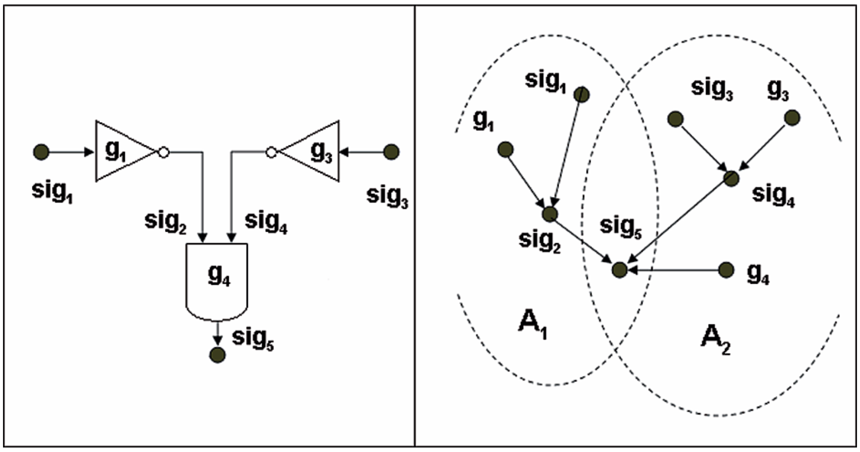


Fig. 11.11. (**a**) A fragment of a digital system. (**b**) Representation of the fragment in subnets of adjacent agents

Next, we consider the directionality of arcs that connect public variables. As mentioned in Sect. 11.4, variables in an agent interface should render the two corresponding agent communities conditionally independent. In other words, no matter whether or not the interface variables have been observed, passing the subjective probability distribution over these variables from one community (through the corresponding agent) should sufficiently inform the other community. When a public variable is involved in a particular type of dependency, termed *induced dependence* (Pearl 1988), it can cause violation of the requirement.

In Fig. 11.11, for instance, the fragment of a digital system in (a) has been partitioned and represented in two agents A_1 and A_2 as shown in (b). The interface between the agents contains variable sig_5 that corresponds to the output signal sig_5 of AND gate g_4. Suppose that agent A_1 observed input signal of NOT gate g_1 to be $sig_1 = 0$ as well as signal $sig_5 = 0$. A_2 observed input signal of NOT gate g_3 to be $sig_3 = 0$. From the intended functions of NOT and AND gates, we know that at least one of g_1, g_3 and g_4 is faulty. However, if A_1 passes its probability distribution on sig_5 to A_2, it is not possible for A_2 to realize that g_3 or g_4 might be faulty, since A_2 has no information about the expected value of sig_2. In fact, A_2 does not even know the existence of variable sig_2 since it is private to A_1.

This problem lies in the fact that none of the agents has all the parent variables of sig_5, namely, sig_2, sig_4 and g_4. To avoid such problem, it is required that every public variable is a *d-sepnode*: A public variable x is a d-sepnode if at least one subnet contains all parent nodes of x from all subnets. The d-sepnode condition can be satisfied in the above example by making variable sig_2 public, as shown in Fig. 11.12.

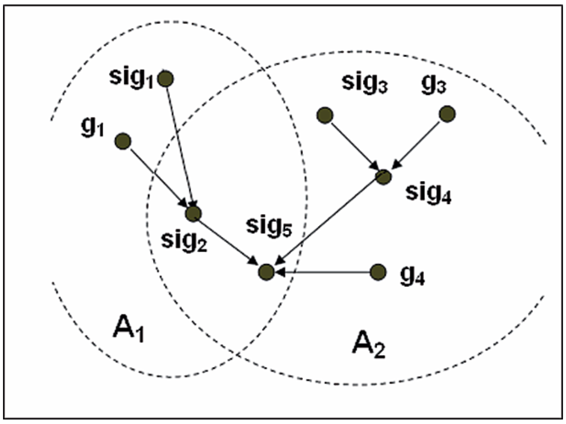


Fig. 11.12. Modified subnet representation that satisfies the d-sepnode condition

With this new subnet representation, A_2 can encode the dependence of sig_5 on sig_2, sig_4 and g_4 within its subnet. If A_1 is to pass a message to A_2, the message not only include the information $sig_5 = 0$, it also include A_1's expectation on the value of sig_2. From the dependency, A_1's expectation on sig_2 (namely, $sig_2 = 1$), and A_2's own expectation on sig_4 (namely, $sig_4 = 1$), A_2 will be able to identify the abnormal behavior of the digital system.

Again, because each subnet is *private* and the parent variables of a public variable may also be private (e.g., the variable sig_2 in Fig. 11.11), the d-sepnode condition cannot be verified by agents working independently. Due to the need to protect agent privacy, it cannot be verified by physically merging individual subnets at a centralizing agent either. A method has been developed (Xiang and Chen 2004) that essentially requires an agent to tell its neighbors, for each of its public variable x, whether it contains any private parent variables of x (but not how many and what they are). Based on these messages, agents can cooperate to detect every non-d-sepnode and to verify every d-sepnode. The method is also implemented in the tool DVerify in WebWeavr toolkit.

To cooperate in verification of global acyclicity and d-sepnode condition, each agent executes a copy of DVerify. One agent, arbitrarily chosen as the coordinator, initiates verification. During verification, messages will be passed among agents along the hypertree, interleaved with local computation at each agent. At the end of the cooperation, the coordinator agent is able to announce whether the MAS has passed the global acyclicity test and d-sepnode test.

11.6 Agent Interface Enhancement

An MSBN-based MAS that has passed the above verification can support autonomous and exact multiagent probabilistic reasoning. However, communication between agents may not be efficient. An agent communicates with an adjacent agent by sending its subjective probability distribution over their interface. For instance, the interface between A_1 and A_3 has 12 binary variables $(e_0, \ldots, e_{11})$ and a message between them contains 4,096 probability values. In general, if the interface consists of m variables and each has k possible values, the message contains k^m probability values.

To reduce the message size while supporting exact inference, factorization of the probability distribution over the interface can be explored. For instance, if variables $e_0, \ldots, e_4$ are conditionally independent of $e_8, \ldots, e_{11}$ given e_5, e_6, e_7, then the message between A_1 and A_3 can be encoded into two distributions over $e_0, \ldots, e_7$ and $e_5, \ldots, e_{11}$ with a total size of $256+128 = 384$: a reduction of factor 10.

How to explore the conditional independence existing in the agent interface through subnet compilation is presented in Sect. 11.7. In this section, we address the situation where no conditional independence relations can be

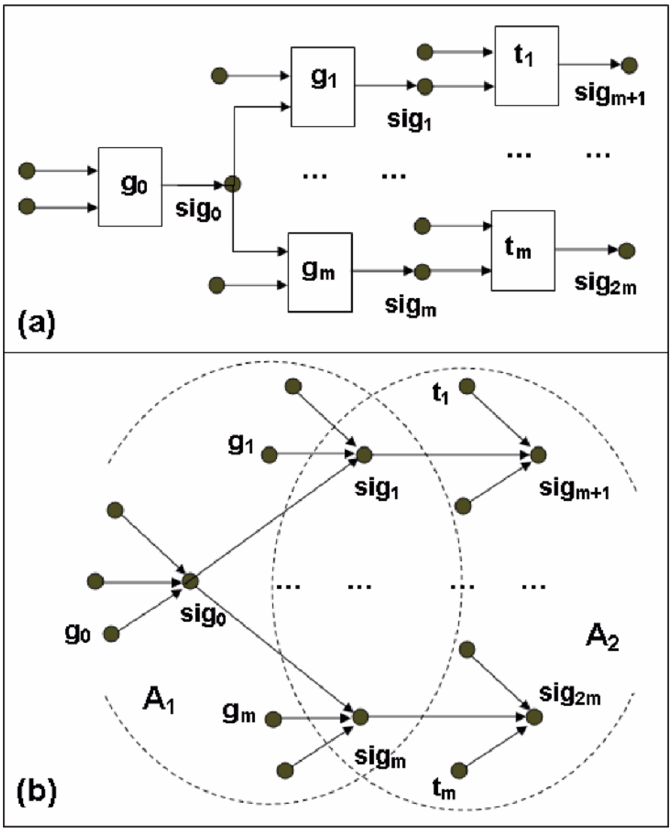


Fig. 11.13. (**a**) A fragment of a digital system. (**b**) Corresponding adjacent subnets

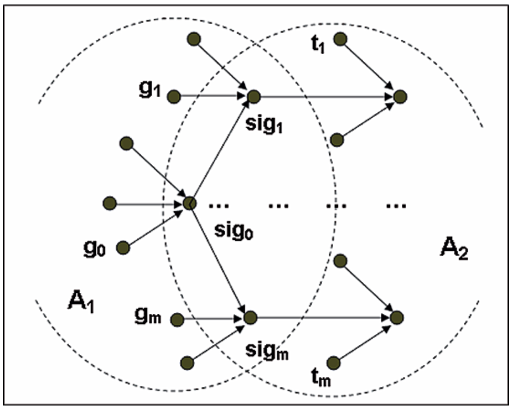


Fig. 11.14. New subnets with enhanced agent interface

found within the natural agent interface or those that exist do not yet offer sufficient efficiency gain. One solution is to enhance the interface with additional variables that can bring conditional independence relations from the subdomain into the interface.

Figure 11.13 illustrates the idea with a fragment of a digital system (a). It is represented as adjacent subnets in (b). The agent interface consists of m variables $\mathrm{sig}_1, \ldots, \mathrm{sig}_\mathrm{m}$. There are no conditional independence relations within the interface. This can be understood as follows: For any three signals sig_i, sig_j and sig_k $(1 \leq \mathrm{i} < \mathrm{j} < \mathrm{k} \leq \mathrm{m})$, if the value of sig_i is known, it helps to generate expectation on the value of sig_0, which in turn generates expectation on the value of sig_j. Even if the value of sig_k is known, it cannot diminish this dependency between sig_i and sig_j. Hence, there are no conditional independence relations among them and a message over the interface has a size of 2^m.

Figure 11.14 shows new subnets where the agent interface is enhanced by adding the variable sig_0. Now, if the value of sig_0 is known, it helps to generate expectation on the value of sig_i. Knowing in addition the value of sig_j cannot change that expectation at all. Hence, sig_i and sig_j are conditionally independent given sig_0: an independence relation has been introduced into the interface. Furthermore, the independence relation holds for every pair of i and j. This allows the probability distribution over the interface to be factorized into m distributions each defined over two variables sig_0 and sig_i $(\mathrm{i} = 1, \ldots, \mathrm{m})$. The total size of the inter-agent message is reduced to 4m from the original size of 2^m.

To explore this idea, suitable variables (such as sig_0) need to be identified. Identification of these variables among a large number of alternatives is nontrivial. The process often requires cooperation between agents. For instance, if $\mathrm{sig}_{\mathrm{m}+1}$ through $\mathrm{sig}_{2\mathrm{m}}$ all feed into a common gate in A_2, then unless its output signal is also added into the agent interface, the above mentioned reduction

Table 11.1. The message size between each pair of adjacent agents before and after interface enhancement

Interface	A0–A1	A1–A2	A1–A3	A3–A4
Before	2,048	1,024	4,096	4,096
After	136	136	336	160

in message size cannot be achieved. Instead of burdening the agent developers with the enhancement task, it can be delegated to agents.

The enhancement involves search through many alternatives, including disclosure of some originally private variables, to neighbor agents, as promising enhancement candidates. To protect agent privacy during enhancement, each agent classifies variables in its subnet into three groups: private, public, and preferably private. The *public* group forms the natural initial agent interface. The *private* group will be kept so absolutely. The *preferably private* group is initially private, but the agent is allowed to make some elements of this group public if it believes that the disclosure may improve efficiency. The agent is required, however, to keep the disclosed variables as fewer as possible. That is, any disclosed candidate variable should be highly promising through the agent's local evaluation. The actual efficiency improvement of an enhancement can only be determined by agents' cooperative evaluation.

A suite of algorithms for multiagent interface enhancement has been developed (Xiang and Zhang 2006). Through multiagent heuristic search, each of the four agent interfaces are enhanced. For example, the interface between A_1 and A_3 (consisting of $\mathrm{e}_0, \ldots, \mathrm{e}_{11}$) is enhanced with additional variables yd_{82}, yd_{101}, yd_{106}, w_{14}, wr_{16}, wr_{18}. These variables bring several independence relations into the interface. For instance, e_θ, e_1, e_2 are independent of e_3, e_4, e_5 given wr_{16}, yd_{106}. As the result of enhancement, the message size between each pair of adjacent agents is reduced significantly, as shown in Table 11.1, with the new message size to be as low as about 4% of the original (between A_3 and A_4).

Agent interface enhancement is the only technical step where information about variables that are initially private (those that are preferably private) may be disclosed. This step is not necessary for exact inference using the MSBN-based MAS and should be regarded as an option for trading privacy with efficiency.

11.7 Compilation into Linked Cluster Trees

Inference computation in an MSBN-based MAS consists of local inference at individual agents and communication among agents. Local inference involves updating the agent's belief (subjective probability distribution) over its subdomain based on local sensor observations. During communication, the basic operation of an agent involves passing to another agent its subjective probability distribution over their interface (the message). The two computations are intertwined: A message for communication must be derived from the sending agent's local distribution over its subdomain, and a message received should be processed for updating the local distribution over the receiving agent's subdomain.

Suppose that an agent's subdomain consists of n variables and each has up to k possible values. The probability distribution over the subdomain has a size of k^n. To make the local inference efficient, the agent must avoid direct manipulation of the distribution. The idea is to explore conditional independence and factorization of the distribution. Each agent compiles its subnet into a cluster tree, where variables are grouped into *clusters* with intersections of adjacent clusters referred to as *separators*. The cluster tree is so constructed such that the intersection of any two clusters is contained in every cluster on the path between them. The property ensures that any update on the probability of a variable located in a cluster can be propagated to every other cluster that contains the same variable. This idea of using such cluster trees for probabilistic inference was proposed first in the single-agent paradigm (see (Lauritzen and Spiegelhalter 1988; Jensen et al. 1990; Jensen 1996; Shafer 1996)). It has been extended into operations under the multiagent paradigm (Xiang 2001; Xiang 2002). Details on compilation can be found from the given reference. Figure 11.15 shows the cluster tree compiled from the subnet of agent A_0 through cooperation with other agents using the tool Structure Compiler in WebWeavr toolkit.

Each cluster is associated with a probability distribution over its member variables obtained from the CPTs in the subnet. The cluster tree encodes the conditional independence relations existing in the subnet: Two adjacent clusters are conditional independent given their separator. These independence relations allow factorization of the the agent's subjective probability distribution over its subdomain. The cluster distributions are more efficient spacewise, yet they uniquely define the agent's subjective probability distribution over its subdomain (Jensen et al. 1990). Furthermore, the tree topology allows local inference to be performed by passing messages (probability distributions) over separators along the tree structure (we describe the inference operation in Sect. 11.8). When the size of the largest cluster is bounded, the inference is efficient.

During communication, an agent needs to send its subjective probability distribution over an agent interface to the neighboring agent. As we discussed earlier, sending the message as a single distribution over the agent interface has the exponential complexity, which motivated agent interface enhancement. However, interface enhancement only ensures that there exist conditional independence relations within the interface. It does not create an explicit data structure to utilize these independence relations. The data structure that serves this purpose is called *linkage tree*.

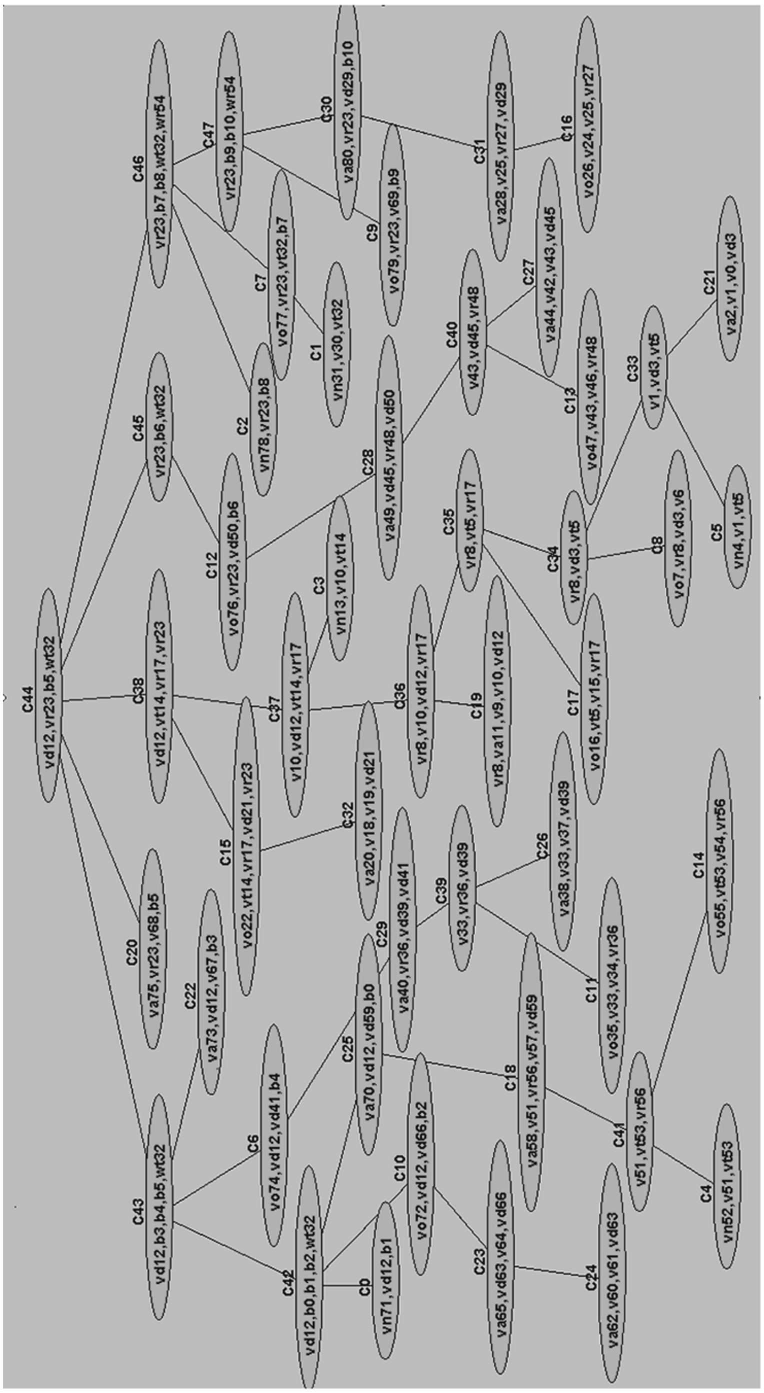


Fig. 11.15. The cluster tree for agent A_0

Fig. 11.16. The linkage tree for computing messages between A_0 and A_1

Essentially, the linkage tree is also a cluster tree. The cluster tree compiled from the agent's subnet is composed of all variables in the agent's subdomain. On the other hand, the linkage tree is composed of only variables in an agent interface and is used only for computation of message to the corresponding adjacent agent. A linkage tree is derived from the local cluster tree and it inherits all the conditional independence relations among interface variables that are explicitly encoded in the local cluster tree. Details on how to derive linkage tree from local cluster tree can be found from reference (Xiang 1996). Figure 11.16 shows the linkage tree of agent A_0 for computing messages to A_1. Each cluster in the linkage tree is called a *linkage*. Each linkage has a corresponding cluster in the cluster tree, called its *host*, that contains the linkage.

Each linkage is associated with a probability distribution that is derived from the distribution associated with its host. From these linkage distributions, the agent's subjective probability distribution over the agent interface can be constructed through factorization. Although the distribution over the interface has a size of $2^{15} = 32768$, the inter-agent message made of linkage distributions has a total size of $3^*2^5 + 2^*2^4 + 1^*2^3 = 136$.

Note that the possibility of efficient message representation using linkage trees is a direct consequence of exploring conditional independence within the agent interface. The compilation operation automatically identifies such independence relations if they exist. If these relations do not yet exist in the natural agent interface, they must be brought into the interface through interface enhancement (Sect. 11.6).

11.8 Multiagent Inference

The above compilation effectively converts the collective knowledge of multiple agents, originally represented as an MSBN, into a set of linked cluster trees. Using the local cluster tree, each agent can perform inference autonomously without cooperation from other agents. For instance, if x is a variable representing a sensor output and an agent observes the value x being logic 1, then the observation can be entered into the cluster tree as follows: First, a cluster that contains x is selected. As mentioned above, the cluster has an associated probability distribution, which specifies the probability for each combination of the values of variables in the cluster. If a combination has the value of x being logic 0, then the probability of the combination is set to 0, meaning that this combination is now impossible given the observation x = logic 1. The remaining combinations will have their probability values scaled up so that they sum to one, while maintaining their original relative magnitudes. This operation is termed *entering observation*.

After each sensor observation has been entered into the corresponding cluster, the change in these clusters must be propagated to other clusters in order to achieve their impact on other variables that depend on them. This is done through message passing along the local cluster tree. Each cluster receives a message from each neighbor cluster. It sends one message to each neighbor after it has received messages from all other neighbor clusters. Each message is simply a probability distribution over the corresponding separator and it is computed from the distribution associated with the sending cluster and messages received by the cluster. Over each separator, exactly two messages are sent, one in each direction.

Since there are no more clusters than variables in the subdomain and there are less separators than clusters, it can be seen that if the clusters are small in size, the message propagation is efficient. It has also been shown (Jensen et al. 1990; Shafer 1996) he cluster probability distribution obtained through the propagation is exact relative to the probability theory, the background knowledge of the subnet, and sensor observations.

Since subnets in our case study represent components that are interconnected and therefore mutually constrained, sometimes communication with other agents allows an agent to better ascertain the current situation of its component than what is achievable by the agent's autonomous inference only. When such is the case, agents engage in a communication operation so that they can benefit from each other's local sensor observations. The communication operates in a similar fashion as the message passing in a cluster tree, but at the agent level and along the hypertree.

Each agent receives a message from each neighbor agent according to the hypertree agent organization. It sends one message to each neighbor agent after it has received messages from all other neighbor agents. A message is a set of probability distributions each of which is over a linkage with the receiving agent and is derived from the probability distribution of the linkage host cluster. Collectively, these distributions define the sending agent's belief over the agent interface. Between each pair of adjacent agents on the hypertree, exactly two messages are sent, one in each direction.

Since there are less linkages than clusters in the local cluster tree, there are as many hypernodes in the hypertree as the number of agents, and there are less hyperlinks than hypernodes, it can be seen that if the clusters are small in size, the agent communication is also efficient. Furthermore, mathematical analysis (Xiang 1996; Xiang 2002) that the agents' beliefs after communication are exact, relative to the probability theory, the collective background knowledge of all agents encoded in their subnets, and sensor observations of all agents. In Sect. 11.9, we demonstrate how these operations can be used to answer the normality and culprit queries.

11.9 Sensor Net Monitoring and Fault Isolation

To monitor the digital system domain, each agent collects sensor outputs and reason about the state of its subdomain autonomously. Less frequently, agents may choose to communicate in order to benefit from information in other agents. Through interleaving local inference and communication, agents can collectively answer the normality and culprit queries.

The tool DMasMsbn in WebWeavr supports agent sensing, inference and communication. We demonstrate digital system monitoring through the following scenario: AND gate wa_{130} in U_1 and OR gate $y0_{49}$ in U_3 are faulty and produce incorrect output signals. The incorrect outputs propagate through other gates and produce more incorrect signals throughout the system. Agents' task is to detect that the system is abnormal (answering the normality query) and to isolate the faulty gates (answering the culprit query).

To demonstrate the operation of the MAS while avoiding the cost of implementing the digital hardware physically, the tool Scenario Simulator from WebWeavr is used to simulate the digital system and associated sensor network. The simulator accepts a set of externally specified input signals to the digital system, simulates the behavior of all digital gates including the faulty gates, and generates output signals of all gates. It responds to agents' request for observations and enforces the assumption that the state of a gate is not observable. When a valid request is received from an agent, the value of the corresponding signal as would be perceived by the sensor will be sent to the agent.

To monitor the domain, each agent is assumed to have the bandwidth to observe at one time as many sensors as about 5% of variables in its subdomain. We assume that all signals are observable except the outputs of the two faulty gates wa_{130} and $y0_{49}$. As there are more observable signals than what are permitted by the bandwidth, some strategy must be utilized to choose what to observe. If gates differ in their prior probabilities of being faulty, those gates with high fault probabilities may be observed with priority. Signals corresponding to their input and output are likely to detect their faults soon after they occur. In the case study, we have assumed the same prior fault probability for all gates. Hence, a random set of signals is observed initially. The first round of observations is shown in Table 11.2.

After entering the observations, each agent updates its belief autonomously. Since these local observations are not sufficient to detect any abnormality within each subdomain, and autonomous reasoning at individual agents cannot take into account the constraints between components, none of the agents detects any problem.

However, after one round of communication among agents, during which one message is passed from each agent to each adjacent agent, the pooling of information allows agents to detect abnormality. A_0 has $P(va_{44} = bad|obs) = 0.025$. Note that this is two and half times higher than the prior fault probability value 0.01. A_1 has a num-ber of gates suspected,

$$wn_{132}, wo_{124}, wo_{163}, wa_{126}, wa_{122}, wa_{139}, wa_{141}, wa_{130},$$

for instance, $P(wa_{130} = bad|obs) = 0.131$. Similarly, A_2 has $P(xa_{32} = bad|obs) = 0.132$, A_3 suspected

$$yn_{39}, yo_{43}, yo_{49}, yo_{15}, yo_{102}, yo_{121}, yo_{95}, ya_{105}, ya_{46},$$

and A_4 suspected zn_{20}, zn_6, zo_{18}, zo_{61}, za_{59}, za_{13}, za_{56}. Therefore, agents have collectively answered the normality query negatively.

The large number of candidate faulty devices is a consequence of propagation of incorrect outputs of the two faulty gates to other devices which causes their outputs to be incorrect. Note that the set of candidates includes the two faulty gates wa_{130} and $y0_{49}$. Therefore, if these devices are replaced, the system will be back to nor-mal. However, that would be too costly. The large number of candidates and low faulty probability value for each tell the agents that further investigation is needed.

Alarmed, each agent makes more observations, subject to the bandwidth restriction. Since the agents now have some candidate gates suspected to be faulty, the observations can be focused on the input and output signals of these gates. A_0 observes signals associated with the suspected gate va_{44}. Its output vd_{45} has been observed. Hence, its inputs v_{42} and v_{43} are observed. After entering observation and autonomous inference, A_0 no longer suspects va_{44}.

Table 11.2. Sensor observations in round 1

A_0	v_9, v_{30}, vr_{23}, vd_{45}, vd_{12}
A_1	w_{37}, wd_{50}, wt_{24}, e_3, w_{89}, w_{136}, w_{121}, w_{120}, w_{119}, w_{107}
A_2	x_{27}, x_9, x_{12}, xd_{33}
A_3	y_{45}, y_{48}, y_{104}, y_{69}, y_{97}, y_{111}, y_{12}, yr_{27}
A_4	z_{60}, z_{58}, z_{55}, z_{15}, z_1, z_{12}

Table 11.3. Sensor observations in round 2

A_0	v_{42}, v_{43}
A_1	wd_{140}, wd_{142}, wt_{133}, wd_{123}, wr_{125}, wd_{127}, c_5
A_2	xd_{16}, c_5
A_3	yd_{47}, yr_{96}, yr_{103}, yd_{106}, yr_{16}, yt_{40}, yr_{44}, e_3
A_4	h_6, zd_{57}, zt_{21}, zr_{19}, zt_7, zd_{14}

A_1 observes the output of each suspected gate, as listed in Table 11.3, except that of wa_{130} (as has been deliberately forbidden to make the decision process more interesting). After entering observation and autonomous inference, A_1 reduces its uncertainty on the original eight candidate faulty gates and now suspects only three:

$$wn_{128}, wn_{132}, wa_{130}.$$

Note that wn_{128} is not one of the gates suspected earlier.

A_2 observes two signals and no longer suspects xa_{32} after inference. A_3 observes 8 signals and decides that $P(yo_{49} = bad|obs) = 0.504$ and $P(yo_{95} = bad|obs) = 0.504$. Given that the signal between the two, yr_{50}, is not observable, this is the best that anyone can achieve. A_4 observes 6 signals. After inference, it decides that its subdomain is normal and does not have any fault.

As A_1 suspects three gates, it makes one more observation related to them: wt_{129}. After inference, it reduces the suspected gates to only wn_{132} (with $P(wn_{132} = bad|obs) = 0.387$) and wa_{130} (with $P(wa_{130} = bad|obs) = 0.617$), which is the best that anyone can achieve given the unobservability of the signal between the two gates.

As the result of the above multiagent inference, A_1 correctly isolates faulty de-vices to wn_{132} and wa_{130}, and A_3 correctly isolates to yo_{49} and yo_{95} (note that wa_{130} and yo_{49} are the true faulty devices). The probability of each of these four devices being faulty is at least 0.387, while all other gates suspected earlier have their probabilities of being faulty dropped to almost zero in all agents. Given that we have forbidden observability between wn_{132} and wa_{130} and between yo_{49} and yo_{95}, the agents have answered the culprit query well. That is, they isolated faulty devices to the smallest possible set given the information available from the sensor network. Replacement of the four devices according to the answer to the query will return the system to its normal state.

What would happen if some agents in the MAS fail? To ensure exactness of inference/communication as well as efficiency, the MSBN-based MAS uses the hypertree agent organization. Because each hyperlink separates the MAS into two agent communities, if the communication link between two adjacent agents fails, the two resultant communities will no long be able to cooperate as we demonstrated above. Furthermore, if an agent of k neighbors fails, the MAS will be broken into k separate communities.

On the other hand, agents in each community can still cooperate within themselves. Theoretical analysis (Xiang 2002) shows that after they communicate, each agent's belief is exact relative to the probability theory, the knowledge encoded in all agents within the community, sensor observations in the entire MAS up to the last communication before the breaking of the MAS, and sensor observations made by all agents in the community since the breaking. Therefore, the MSBN framework allows the MAS to fail gracefully, rather than to function as all or nothing.

11.10 Summary

MSBNs extend BNs to provide a rigorous computational framework for intelligent sensor network applications. The key advantages of the framework are the following:

1. Agents' beliefs regarding the interpretation of the sensor observations are exact according to Bayesian probability theory.
2. Inference at each agent is autonomous and no centralized control is needed.
3. Communication within the agent society can be initiated by any agent and no fixed controller is needed.
4. As long as the dependence structure of agents are sparse, the inference and communication are efficient.
5. Operations for model construction, model compilation, inference, and communication protect agent privacy. Agent interface enhancement is the only step where information about preferably private variables may be disclosed. This step is not necessary for exact inference using the MSBN-based MAS and should be regarded as an option for trading privacy with efficiency.

The performance guarantees (on autonomy, exactness, efficiency and privacy) offered by the MSBN framework require careful model construction, model compilation and inference-communication operations. Most of the compilation, inference and communication operations can be fully automated as demonstrated by tools in WebWeavr toolkit. The model construction is the step that demands particular effort from sensor network practitioners even with the aid of tools. The modeling task can be broken down into the following:

1. The integrator of the MAS needs to partition the domain into subdomains over which individual agents will be developed, to specify agent interfaces, and to define the hypertree agent organization. For many problems, there exists some natural partition. Care must be taken so that all requirements on agent interfaces are satisfied.
2. The developer of each agent must specify the agent's subnet. This includes the dependence structure over the agent's subdomain in terms of a DAG and the CPT for each node in the DAG.
3. Once the agent organization and subnets are specified, they should be subject to verification. If global acyclicity and d-sepnode conditions are violated, the subnets must be revised. Negotiation among agent developers and integrator is needed to determine alternative modifications to subnets and who will make the changes.
4. The modeling task may not end yet after subnets pass verification. After they are compiled into linked cluster trees, resultant linkage trees may not support efficient communication. In such case, interface enhancement is needed. To enable enhancement, for each subnet, the agent developer needs to specify a subset of variables as preferably private so that they may be added to the agent interface.

5. Once agent interface is sufficiently efficient, measured by the size of the largest linkage, the model construction is complete.

As any model of a complex domain, it only reflects the best knowledge available at the time. As new information becomes available, the model may be refined. The functional units of an MSBN-based MAS are the agents. Therefore, the modeling units are the subnets embodied by agents. A subnet consists of its graphical structure and its numerical CPTs. When the problem domain is an artifact, such as a piece of equipment, the subnet structure is constructed from the structure of the designed artifact. Hence, unless the artifact is modified, it is unlikely that the subnet structure needs refinement. On the other hand, CPTs encodes information on the artifact's faulty behavior, which is not designed. Therefore, refinement of CPTs is not only possible but also desirable. It can be easily accomplished by utilizing the fault frequency data accumulated. As subnets are thus refined, the MAS will perform more effectively.

Acknowledgements

The financial support from National Sciences and Engineering Research Council (NSERC) of Canada through Discovery Grant is acknowledged.

References

D.S. Bernstein, S. Zilberstein, and N. Immerman. The complexity of decentralized control of Markov decision processes. In Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, Stanford, 2000, pages 32–37

C. Boutilier. Multiagent systems: Challenges and opportunities for decision-theoretic planning. AI Magazine, 35–43, Winter 1999

E. Castillo, J. Gutierrez, and A. Hadi. Expert Systems and Probabilistic Network Models. Springer, Berlin Heidelberg New York, 1997

R.G. Cowell, A.P. Dawid, S.L. Lauritzen, and D.J. Spiegelhalter. Probabilistic Networks and Expert Systems. Springer, Berlin Heidelberg New York, 1999

R.T. Cox. Probability, frequency and reasonable expectation. American Journal of Physics, 14(1):1–13, 1946

J. Doyle. A truth maintenance system. Artificial Intelligence, 3(12):231–271, 1979

de Finetti. Foresight: its logical laws, its subjective sources. Annales de I'Institut Henri Poincare, 7:1–68, 1937. Reprinted in 1980 in Studies in Subjective Probability (H.E. Kyburg and H.E. Smokler, Eds.), 93–158

C. Guestrin, P. Bodi, R. Thibau, M. Paski, and S. Madde. Distributed regression: An efficient framework for modeling sensor network data. In Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks, Berkeley, CA, 2004, pages 1–10

M.N. Huhns and D.M. Bridgeland. Multiagent truth maintenance. IEEE Transactions On Systems, Man, and Cybernetics, 21(6):1437–1445, 1991

F.V. Jensen. An Introduction To Bayesian Networks. UCL Press, 1996
F.V. Jensen. Bayesian Networks and Decision Graphs. Springer, Berlin Heidelberg New York, 2001
F.V. Jensen, S.L. Lauritzen, and K.G. Olesen. Bayesian updating in causal probabilistic networks by local computations. Computational Statistics Quarterly, (4):269–282, 1990
S.L. Lauritzen and D.J. Spiegelhalter. Local computation with probabilities on graphical structures and their application to expert systems. Journal of Royal Statistical Society, Series B, (50):157–244, 1988
C.L. Mason and R.R. Johnson. DATMS: A framework for distributed assumption based reasoning. In L. Gasser and M.N. Huhns, editors, Distributed Artificial Intelligence II, pages 293–317, Pitman, London, 1989
J. McCarthy. Circumscription – A form of non-monotonic reasoning. Artificial Intelligence, 13:27–39, 1980
D.V. McDermott and J. Doyle. Non-monotontc logic i. Artificial Intelligence, 13:41–72, 1980
S. Moral, R. Rumi, and A. Salmeron. Mixtures of truncated exponentials in hybrid Bayesian networks. In Lecture Notes in Artificial Intelligent, volume 2143, pages 135–143. Springer, Berlin Heidelberg New York, 2001
R. Nair, P. Varakantham, M. Tambe, and M. Yokoo. Networked distributed POMDPs: A synthesis of distributed constraint optimization and pomdps. In Proceedings of the 20th National Conference on Artificial Intelligence, 2005, pages 133–139
R.E. Neapolitan. Probabilistic Reasoning in Expert Systems. Wiley, New York, 1990
H.P. Nii. Blackboard systems: The blackboard model of problem solving and the evolution of blackboard architectures. AI Magazine, 7(2):38–53, 1986
J. Pearl. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Francisco, 1988
A. Rao and M. Georgeff. Deliberation and its role in the formation of intentions. In B. D'Ambrosio, P. Smets, and P.P. Bonissone, editors, Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence, pages 300–307, Morgan Kaufmann, San Francisco, 1991
R. Reiter. A logic for default reasoning. Artificial Intelligence, 13:81–132, 1980
N. Roos, A.T. Teije, and C. Witteveen. A protocol for multi-agent diagnosis with spatially distributed knowledge. In Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems, Melbourne, 2003, pages 655–661
S. Russell and P. Norvig. Artificial Intelligence: A Modern Approach. Prentice Hall, Englewood Cliffs, NJ, 2003
G. Shafer. Probabilistic Expert Systems. Society for Industrial and Applied Mathematics, Philadelphia, 1996
M. Wooldridge. An Introduction to Multiagent Systems. Wiley, New York, 2002
Y. Xiang. A probabilistic framework for cooperative multi-agent distributed interpretation and optimization of communication. Artificial Intelligence, 87(1–2): 295–342, 1996
Y. Xiang. Verification of dag structures in cooperative belief network based multi-agent systems. Networks, 31:183–191, 1998
Y. Xiang. Cooperative triangulation in MSBNs without revealing subnet structures. Networks, 37(1):53–65, 2001

Y. Xiang. Probabilistic Reasoning in Multi-Agent Systems: A Graphical Models Approach. Cambridge University Press, Cambridge, UK, 2002

Y. Xiang and X. Chen. Interface verification for multagent probabilistic inference. In J.A. Gamez, S. Moral, and A. Salmeron, editors, Advances in Bayesian Networks, pages 19–38. Springer, Berlin Heidelberg New York, 2004

Y. Xiang and V. Lesser. On the role of multiply sectioned Bayesian networks to cooperative multiagent systems. IEEE Transactions on Systems, Man, and Cybernetics-Part A, 33(4):489–501, 2003

Y. Xiang and K. Zhang. Agent interface enhancement: Making multiagent graphical models accessible. In Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'06), pages 19–6, 2006

P. Xuan, V. Lesser, and S. Zilberstein. Communication in multi-agent Markov decision processes. In Proceedings of the 6th International Conference on Multi-agent Systems, 2000, pages 467–468

Y. Xiang. WebWeavr-IV Research Toolkit. http://www.cis.uoguelph.ca/yxiang/

12

An Intelligent Expert Systems' Approach to Layout Decision Analysis and Design under Uncertainty

Abdul-Rahim Ahmad[1], Otman Basir[1], Khaled Hassanein[2], and Shahid Azam[3]

[1] Systems Design Engineering, University of Waterloo, Canada {arahim, obasir}@engmail.uwaterloo.ca
[2] School of Business, McMaster University, Hamilton, Canada hassank@mcmaster.ca
[3] Environmental Systems Engineering, University of Regina, Canada Shahid.azam@uregina.ca

Summary. This chapter describes an intelligent soft computing based approach to layout decision analysis and design. The solution methodology involves the use of heuristics, metaheuristics, human intuition as well as soft computing tools like artificial neural networks, fuzzy logic, and expert systems. The research framework and prototype contribute to the field of intelligent decision making in layout analysis and design by enabling explicit representation of experts' knowledge, formal modeling of fuzzy user preferences, and swift generation/manipulation of superior layout alternatives to facilitate the cognitive, ergonomic, and economic efficiency of layout designers.

12.1 Introduction

The Layout Design (LD) process is geared towards seeking some superior outcome in the spatial arrangement of modules in a given space while satisfying a set of given preferences and constraints. A generic approach to the LD problem is to treat it as an oriented and orthogonal two-dimensional rectangular packing problem (2D-BPP). In this problem, n rectangular modules of length L_i and width W_i $(i = 1, 2, \ldots, n)$ are to be packed on a large rectangular packing space of length L_o and width W_o without overlaps and within the boundary constraints (Dyckhoff 1990, Garey and Johnson 1979). Each module i is of fixed orientation and must be packed with its edges parallel to the edges of rectangular packing space. Each module i is associated with a utility u_i and the objective is to maximize the total utility of the packing pattern. This problem is relevant to various facilities planning, cutting, packing, storing, transporting, scheduling, and resource allocation functions of businesses

A.-R. Ahmad et al.: *An Intelligent Expert Systems' Approach to Layout Decision Analysis and Design under Uncertainty*, Studies in Computational Intelligence (SCI) **97**, 321–364 (2008)
www.springerlink.com

(Islier 1998, Lodi et al. 2002, Martens 2004). Only in facilities planning area, US businesses spend about a trillion dollars in new facilities annually and their layouts directly affect more than 20% of the operating costs (Ahmad 2005). Thus, the research efforts in improving the efficiency and efficacy of tools for layout decision analysts and decision-makers are imperative and ongoing.

The LD is a tedious process that calls for sophisticated decision analysis and design support. The existing solution approaches largely employ very rigid and overly simplistic design algorithms and guidelines, largely without an elaborate methodology for their utilization. Nevertheless, the complex, subjective, uncertain, and evolving nature of layout design preferences and fitness objectives means that the synergistic use of available modeling and design tools as well as an expertise in tradeoffs lies at the heart of any layout design and analysis process. Consequently, any good automated layout design system should be flexible and robust enough to facilitate adaptation to the evolving scenarios as well as incorporation of cognitive and sub-cognitive expertise of domain experts. However, most traditional approaches to the LD problem lack the requisite flexibility, efficacy, and robustness, as discussed in detail in the subsequent sections (Abdinnour-Helm and Hadley 2000, Ahmad 2005, Badiru and Arif 1996, Osman et al. 2003). The situation is further complicated by the high cognitive overhead encountered by layout designers in acquiring, remembering, understanding, and applying the vast body of subjective and uncertain information/preferences available to them.

Recent developments in the field of intelligent systems have rendered powerful soft computing tools for tackling with such complex and uncertain problems as layout design. Such alternatives include an array of emerging computing disciplines such as Decision Support Systems, Expert Systems, Fuzzy Logic, Neural Networks, Genetic Algorithms, and hybrids like Neuro-Fuzzy-Genetic systems (Ahmad 2005, Karray and De Silva 2004). These technologies share the common denominator in their digression from classical reasoning and modeling approaches through a set of more flexible computing tools (Negnevitsky 2002). Such approaches are gaining favor in modeling cognition and intelligent systems, as the underlying procedures are most analogous to human reasoning (Ahmad 2002, Akoumianakis et al. 2000, Zadeh 1999). Such technologies have demonstrated the power and philosophy to solve complex and ill-defined problems, offering significant potential in dealing with the LD problem.

In this chapter, a promising research framework for an Intelligent System for Decision Support and Expert Analysis in Layout Design (IDEAL) is presented. The research framework is aimed at addressing some of the major issues involved in using the sub-cognitive, subjective, and fuzzy design preferences as a key to enhancing productivity of layout designers. Instead of pursuing some perfect methods, our emphasis is on the development of a generic research paradigm and a tool that could be used in furthering the research in layout planning by supplementing the knowledge, experience, and design intuition of layout planners. Our approach involves tackling various important aspects of the problem through a synergistic utilization of some promising soft computing techniques, advanced heuristics, and metaheuristics.

The rest of the chapter is organized as follows. Section 12.1 provides motivation for our research. Section 12.2 presents a brief literature review of some relevant faculties and their significance in this research. Section 12.3 provides an overview of traditional approaches to the LD problem. Section 12.4 provides a survey of intelligent and knowledge-based approaches to the LD problem. Section 12.5 delineates the proposed solution paradigm and its various major constituents. Section 12.6 outlines results of some case studies undertaken to test the effectiveness of the proposed paradigm. Section 12.7 lists some promising research directions. Section 12.8 concludes the chapter.

12.2 Literature Survey

The diverse scope of the LD problem means that a substantial literature is available in a variety of work domains (Abdinnour-Helm and Hadley 2000, Ahmad 2005, Akoumianakis et al. 2000, Burke et al. 2004, Karray et al. 2000, Tompkins et al. 2002, Youssef et al. 2003). This problem has been variously referred to as *topology optimization* (Mir and Imam 1992), *block placement* (Ahmad 2005), *macro cell placement* or *VLSI layout design* (Schnecke and Vonberger 1997), *layout optimization* (Cohoon et al. 1991), *facilities layout* (Tam et al. 2002), *plant layout* or *machine layout* (Hassan and Hogg 1994), *bin-packing* (Jakobs 1996), *partitioning* (Moon and Kim 1998), etc. However, we may classify LD problems into four major application categories including Facilities LD, Circuit LD, User Interface LD, and Cutting/Packing. A brief description of the significance and prevalence of the LD problem within these contexts is provided here.

In facilities LD, various activities and components are allocated spaces in the given periphery (Abdinnour-Helm and Hadley 2000). The resulting layout of facility establishes the physical relationship among activities and their objectives (Badiru and Arif 1996, Welgama et al. 1995). It may also have profound effects on such relatively intangible matters as environment and safety. Consequently, these space allocation decisions are based on various commutation, communication, political, social, environmental, and safety considerations (Meller and Gau 1996). Indeed, an adequately designed facility layout improves the efficiency, efficacy, productivity, and profitability of an organization (Norman and Smith 2002). The relative permanency of outcome and the scale of strategic investment stipulations mean more research efforts have been dedicated to facility LD than any other LD area.

The bin-packing problem is directed at packing a greater number of items in the smallest number of fixed size bins (Dyckhoff 1990). As such, the typical goal is to maximize the space utilization (Kim et al. 2001). Among the several variants of general bin-packing problem, we limit ourselves to the oriented and orthogonal two-dimensional rectangular packing problem (2D-BPP). This problem provides a basis for devising a generic approach to 2-D layout design and used for elaborating our research paradigm.

The design of VLSI microchips involves several phases including functional design, circuit design, physical design, and fabrication (Mazumder and Rudnick 1999). An important step in physical design is the macrocell placement based on a range of subjective and conflicting preferences and constraints (Moon and Kim 1998). Macrocells are the circuit components lumped together in functional entities with connection terminals along their borders. These terminals are connected by signal nets, along which signals or power is transmitted among the various components. As such, the macrocell layout also characterizes routes selected for the signal nets. During the macrocell placement phase, an estimated amount of routing space or white space is added between the cells.

12.2.1 Popular Approaches to Mathematical Formulations

A range of formulations for the LD problem has been proposed in the literature and a good account of these can be found in (Ahmad 2005, Bozer and Meller 1997). The most popular of such formulations include the *Quadratic Assignment Problem* or QAP (Bazaraa 1975), the *Quadratic Set-Covering* problem or QSC (Bazaraa 1975), and the *Two-Dimensional Bin-Packing Problem* or 2D-BPP (Ahmad 2005).

QAP formulations deal with decisions regarding location of equal area modules. This approach works by assigning one module to every location and at most one module to a given location. Due to NP-Complete nature, it is very hard to procure a verifiably optimal solution for more than 16 modules (Meller and Gau 1996).

QSC formulation requires data on the size of each module, candidate locations of each module, and utilities of each module. QSC allows layout designers to introduce candidate locations of each modules, which helps in eliminating undesirable placements. It also takes the advantage of the intuition and expertise of the user, while reducing computational efforts by restricting the search space. Nevertheless, QSC requires a large number of user inputs for every module under consideration (Bazaraa 1975, Ligget 2000).

The LD problem may also be formulated as an oriented and orthogonal 2D-BPP. It has the advantage of maintaining the integrity and the shape of modules. Such a formulation requires minimal post-optimization processing in comparison with other prevailing LD problem formulations. Furthermore, it constitutes a generic approach to many LD problems (Ahmad 2005, Burke et al. 2004, Dyckhoff 1990, Garey and Johnson 1979, Lodi et al. 2002).

Existing mathematical formulations of LD problem have substantial limitations that make these formulations somewhat incompatible with most real world applications. For instance, the QAP does not allow control over the shape of modules in the resulting layout and QSC requires a large number of user inputs for every module under consideration (Deb and Bhattacharyya

2004). These mathematical models offer little practical advantage in dealing with real layouts of any consequence due to the prohibitive size of the associated mathematical program. Such core issues as ill-structured, subjective and uncertain character of the layout preferences further exacerbate the situation (Malakooti and Tsurushima 1989). In addition, such mathematical programs rely on crisp values of various parameters that are, presumably, measured accurately and attributed to specific dynamics of the problem (Irani and Huang 2000, Mir and Imam 2001). In reality, such data is often available only for some unrealistically simplified layout planning scenarios. Consequently, these formulations are of little practical advantage when a modestly large size problem, involving subjective and uncertain preferences, is considered. Consequently, fast and efficient heuristics that consistently provide superior solutions are the major focus in this area (Burke et al. 2004).

12.3 Traditional Solution Approaches

Various heuristic and analytical techniques have been published for finding solutions to the LD problem. The heuristic techniques find solutions to the problem mostly by treating it as a QAP (Bazaraa 1975, Wu et al. 2002). The 2-dimensional plane is discretized into a grid structure, which results in high computational costs (Gloria et al. 1994). Other solution approaches include tree search algorithms (Pierce and Crowston 1971), binary mixed integer-programming (Love and Wong 1976), and network decomposition (Mak et al. 1998) etc. The NP-Hard and subjective nature of the LD problem means that traditional hard optimization approaches do not hold much promise. Nevertheless, a significant body of research is available in this area. Here we briefly discuss some existing traditional approaches to the LD problem with an emphasis on their limitations.

12.3.1 Algorithmic Approaches

Here we discuss some popular algorithmic approaches to solving layout design problem.

The development of a layout through a *Graph* based approach involves three main steps. First, developing an adjacency graph using inter-module interactions of adjacent pairs of modules. Second, constructing the dual graph of the adjacency graph. Third, converting the dual graph to a block layout specifying actual shapes and areas of modules. It should be noted that the combinatorial nature of the number of arcs in the second step makes the problem particularly difficult to solve. It implies that some heuristics must be employed to limit the number of arc incidents on each module. In addition, similar to the QAP approach, even a small size problem involving non-identical modules cannot be solved with guaranteed optimal solution. Detailed review of such graph-search approaches and heuristics can be found in the literature (Foulds 1995, Hassan and Hogg 1994).

Tree Search methods are more relevant to constraint satisfaction style formulation of the LD problem (Hower 1997). Such search mechanisms incrementally construct layout solutions by adding one module at a time to a partial layout while testing for any violation of feasibility constraints. A tree search method may employ a either breadth-first search by enumerating all possible ways of adding a new module or depth-first search by creating a full layout by placing all the modules sequentially (Akin et al. 1997). However, such an approach is inherently inefficient and requires frequent backtracking when feasibility constraints are violated, which adds to the computational complexity (Ligget 2000).

There are various *analytical techniques* dealing with continuous design space with relatively minimal computational requirements (Adya et al. 2003, Mir and Imam (1992, 1996, 2001), Tam et al. 2002, Welgama et al. 1995). However, analytical approaches have yet to be developed to furnish results comparable to advanced heuristic/metaheuristic techniques. Nevertheless, these provide more insights to the structure of the problem leading to advanced and effective heuristics.

12.3.2 Metaheuristic Approaches

Decision-makers often resort to heuristics for dealing with difficult and uncertain problems. Similarly, the NP-Hard and subjective nature of the LD problem suggests that heuristics can be very effective in solving the problem. Accordingly, various heuristic algorithms for solving the difficult 2D-BPP are available in the literature (Ahmad 2005, Dowsland et al. 2002, El-Bouri et al. 1994, Hopper and Turton 2001, Jakobs 1996, Kim et al. 2001, Leung et al. 2003, Liu and Teng 1999, Lodi et al. (1999, 2002), Martens 2004). In this regard, the importance of effectively limiting an intractable search space to some reasonable subset of possible solution topologies cannot be overemphasized (Dowsland et al. 2002, Tompkins et al. 2002). Understandably, several effective metaheuristic solution methodologies are proposed in the literature. The core of such approaches is quite simple and involves treating the LD problem as a packing problem by defining an *ordering of modules* in the form of a sequence or permutation and a *placement* or *decoding heuristic* for placing modules in the determined order (Ahmad 2005, Leung et al. 2003). Recent metaheuristics that have shown good results for LD include simulated annealing (Adya et al. 2003), genetic algorithms (Ahmad 2005, Gloria et al. 1994, Martens 2004), tabu search (Hopper and Turton 2001), random search (Ahmad 2005, Jakobs 1996, Liu and Teng 1999), naive evolution Hopper and Turton 2001, and hybrids (Ahmad 2005, Lee and Lee 2002). The key to these methods generally lies in some effective means for getting out of local minima. However, the speed and effectiveness of such metaheuristic approaches are largely determined by the speed and effectiveness of decoding heuristics (Hopper and Turton 2001).

Earlier research on the relative performance of some of these popular metaheuristics in solving the LD problem, at best, provides mixed results (Hopper and Turton 2001, Leung et al. 2003, Youssef et al. 2003). Nevertheless, some knowledge of the merits and the demerits of these metaheuristic approaches, within the context of the LD problem, could result in a more judicious selection of optimization method. Consequently, here we discuss some merits and demerits to provide some insights to these popular metaheuristics.

Genetic Algorithms (GA) are primarily used due to the non-deterministic and global optimization approach that has the potential to provide several near optimal and diverse layout alternatives (Ahmad et al. 2006, Youssef et al. 2003). GA allow incorporation of domain-specific knowledge into the fitness of individual solutions as well as in genetic selections and operations (Youssef et al. 2003). Moreover, GA creates a population of optimized solutions.

GA have been applied to the LD problem in various ways. However, much of the research deals with relatively simple problems requiring assignment of identical modules to given locations. Comparative studies of GA with other metaheuristics show superiority of GA in LD (Hopper and Turton 2001). As such, GA provide a very promising approach for LD through generation of a diverse set of superior alternatives (Ahmad 2005, Lee and Lee 2002, Martens 2004, Moon and Kim 1998). Further advantages of GA within the context of LD are discussed in Sect. 12.5.1.

Simulated Annealing (SA) is a well-known, high-performance, and effective stochastic optimization technique for combinatorial problems (Mir and Imam 2001, Tam et al. 2002). Any domain specific knowledge is incorporated mainly in the SA cost function (Youssef et al. 2003). SA starts with a random solution and makes incremental refinements by moving genes from their current location to new locations, generating new solutions. Moves that decrease the cost are accepted while moves that increase the cost are also accepted with a probability that decreases exponentially with time. Thus, it avoids being trapped in a local optimum by accepting inferior solutions, too.

SA is known to be a stable metaheuristic approach capable of finding a global optimal solution (Youssef et al. 2003). However, SA is generally very slow to converge to good solutions when compared to GA. SA may provide solutions to an LD problem that is comparable to or marginally better than GA (Hopper and Turton 2001, Youssef et al. 2003). The downside is that SA operates on only one solution at a time and has a meager history or memory for learning from past explorations. In short, SA can be characterized as a serial algorithm that is not easily amenable to parallel processing without significant communications overhead. Another implication is the production of closely related solutions, eluding the requirement of having both superior and diverse layout alternatives (Ahmad 2005).

Tabu Search (TS) is another successful, effective, and robust metaheuristic approach for solving complex combinatorial and continuous optimization problems. In a generic sense, TS is an iterative procedure that starts from some initial feasible solution and attempts to determine a better solution by making several neighborhood moves. The set of admissible solutions explored at a particular iteration forms a candidate list and TS selects the best solution from the candidate list.

A distinguishing feature of TS is its exploitation of an adaptive and explicit form of memory in the shape of a tabu list, which is used to prevent back cycling and influence the search (Youssef et al. 2003). The tabu list is analogous to a window on accepted moves that permit the search beyond the points of local optimality while making the best possible move.

Naive Evolution (NE) search is somewhat similar to GA in its basic form. However, it employs only a mutation operator in order to generate successive populations of solutions. Understandably, it is very easy to implement but lacks the structured search engendered by crossover operators in GA. The complexity and subjectivity involved in most LD applications mean that the even NE may turn out to be an effective and efficient search strategy (Hopper and Turton 2001).

Random Search (RS) is another naive search strategy where the ordering of modules is generated randomly (Ahmad 2005, Ahmad et al. 2006, Hopper and Turton 2001). Again, the subjectivity and complexity in most LD applications mean that an RS strategy could result in quite superior outcomes. However, the superiority of such solutions does not match to those generated by such advanced metaheuristics as SA and GA (Youssef et al. 2003).

12.3.3 Heuristic Approaches

The combinatorial complexity of the LD problem formulations has lead to development of various efficient heuristics, which may be used alone or in conjunction with metaheuristics. Indeed, metaheuristics based solution approaches to the LD problem require effective and efficient placement or decoding heuristics for determining the physical position of modules in the resulting layout configuration. In effect, a module placement algorithm takes one module at a time from a sequence of modules and determines its position in the packing space based on pre-specified steps, usually designed to realize some local improvements in the search process (Healy et al. 1999, Youssef et al. 2003). An efficient placement strategy that generates superior layouts is critical for the efficacy of such an endeavor (Dowsland et al. 2002). Here we discuss some of the most efficient, effective, and documented decoding heuristics, namely Bottom-Left, Improved Bottom-Left, and Bottom-Left Fill (Burke et al. 2004, Dowsland et al. 2002, Hopper and Turton 2001). In Sect. 12.5.1, we provide some a new decoding heuristic and demonstrate its efficiency and efficacy.

The *Bottom-Left* (BL) placement algorithm calls for placing a module at the bottom-most and left-most feasible position through successive vertical and horizontal movements of the module (Ahmad et al. 2006, Chazelle 1983, Dowsland et al. 2002, Healy et al. 1999, Hopper and Turton 2001, Jakobs 1996, Liu and Teng 1999). Starting from the top-right corner of the packing space, each module is pushed as far as possible to the bottom and then as far as possible to the left (Jakobs 1996). The apparent advantages of such approaches include speed and simplicity (Dowsland et al. 2002). However, BL tends to leave holes in the packing rendering poor space utilization.

Various improvement schemes have been proposed for the BL such as the *Improved-BL heuristic* (IBL) (Liu and Teng 1999). Such improved strategies give precedence to a shift towards the bottom and allow module rotations. However, even these improvised strategies encounter such problems as dead-area and inferior aesthetic contents.

The *Bottom-Left Fill* (BLF) placement algorithm is a more sophisticated version of BL, attempting to fill empty spaces by placing a module into the lowest available position and maintaining a list of candidate placement locations. Consequently, BLF overcomes the problem of poor space utilization. Nevertheless, the major disadvantage lies in its $O(n^\wedge 3)$ time complexity (Burke et al. 2004, Chazelle 1983, Hopper and Turton 2001).

The BL and the IBL are overly simplistic heuristics with inherent deficiencies such as poor space utilization. The optimal packing configuration may be obtained by the BL even after exhaustive enumeration (Jakobs 1996). In addition, the BL, the IBL, and the BLF are not very effective in incorporating qualitative considerations such as the layout symmetry and aesthetics. Further, these algorithms are more appropriate for the minimization of the packing height. Consequently, the quest for efficient and effective module placement strategies is an interesting and popular research direction (Burke et al. 2004).

12.4 Intelligent and Knowledge-Based Approaches

Intelligent and knowledge-based approaches are very promising in the LD area. Here we provide a discussion on the promise of these approaches.

12.4.1 Decision Support Systems

Incidentally, the layout design is not an exact science. Indeed, it is irrational to expect that a specific layout would surpass all others for every evaluation objective (Turban and Aronson 2001). Consequently, the generation of superior layout alternatives in a flexible and automated manner is critical to any LD process (Turban and Aronson 2001). Conceivably, some DSS mechanism could be beneficial in solving the LD problem.

Decision Support Systems (DSS) represent a class of computerized information systems that utilize the knowledge about a specific application domain to assist decision makers by recommending appropriate actions and strategies (Turban and Aronson 2001). The DSS problem-solving paradigm provides a means for assisting decision makers in retrieving, summarizing, and analyzing decision relevant data. Consequently, it results in a reduction in the cognitive overload faced by the decision maker(s). Research has shown that DSS techniques are useful in generating and evaluating a large number of alternative solutions and effectively helping decision-makers in arriving at better decisions (Turban and Aronson 2001). Some research can found in the literature that attempts to solve the problem through the DSS paradigm. Here we describe a couple of such systems.

Foulds (1995) describes a system called LayoutManager that is reportedly deemed a decision support system in facilities planning. LayoutManager permits users to select the layout design algorithm and other necessary starting conditions. The problem specific data must be provided in a standard format through a text file. Any modifications to the design parameters require direct editing of this text file. In order to generate a layout alternative, user selects a starting module, a graph search heuristic, and a rigid fitness metric. Further alternatives may be generated through trial and error. The deterministic layout design heuristics, based on graph search, do not allow diversified and extensive search of the solution space. The LayoutManager does not provide any means for giving users any real control over the proceedings. Furthermore, it does not provide functionalities that would allow users to interactively make any informed or knowledge-based interventions or even manipulations of the layout alternatives produced by the system. In short, the system lacks the flexibility, efficiency, efficacy, scalability, and robustness that would be logical requisites for a DSS in LD.

Tam et al. 2002 describe a nonstructural fuzzy decision support system (NSF-DSS) that integrates both experts' judgment and computer decision modeling, making it suitable for the appraisal of complicated construction problems. The system allows assessments based on pairwise comparisons of alternatives. However, this pairwise comparison approach is inherently inefficient and requires frequent and expensive backtracking. Nevertheless, the research reported in Tommelein 1997 provides many useful insights and future research directions in this field.

12.4.2 Expert Systems

Incidentally, the layout design is not an exact science. Indeed, it is irrational to expect that a specific layout would surpass all others for every evaluation objective (Tommelein 1997). Consequently, the generation of superior layout alternatives in a flexible and automated manner is critical to any layout planning process. Conceivably, some DSS mechanism could be beneficial in solving the LD problem.

An Expert System (ES) is defined as an intelligent computer program that applies reasoning methodologies or the knowledge in a specific domain to render advice or recommendations – much like a human expert (Tompkins et al. 2002) ES are usually characterized by the existence of a large repository of knowledge for solving problems in a very constricted work domain (Malakooti and Tsurushima 1989). Such a knowledge repository may comprise of human

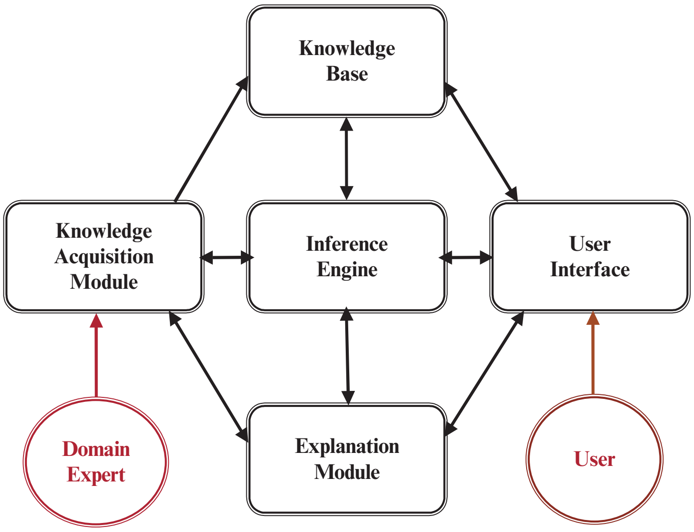


Fig. 12.1. A typical expert system

knowledge and expertise formulated as specific rules and heuristics (Jackson 1999). The distinguishing feature between ES and DSS is the separation of knowledge and the reasoning method involved in an ES, resulting in greater modularity in the system (Negnevitsky 2002). As such, ES afford a greater degree of flexibility, thus making it the paradigm of choice for our research in automating the LD process. Furthermore, ES provide explanation capability as a mean of understanding the reasoning behind a decision.

A traditional ES is shown in Fig. 12.1. It has five basic components, namely a Knowledge Acquisition Module, a Knowledge Base, an Inference Engine, an Explanation Facility, and an interactive User Interface (Negnevitsky 2002). The details about individual components and their synergy follow in Sect. 12.5 within the context of the proposed intelligent system for decision support and expert analysis in layout design. An ES designed specifically to aid decision makers continuously increases productivity, lowers costs, and spurs innovation (Ahmad 2005). However, existing literature on the application of the ES paradigm in LD is quite meager. In addition, such systems have considerable shortcomings, summarized as follows.

Fisher and Nof (1984) present a FAcilities Design Expert System (FADES) for machine LD applications. The reported prototype contains various heuristics and an inferencing mechanism to select a heuristic appropriate for the given scenario. Knowledge is represented using first-order predicate logic. FADES can only solve small-scale problems consisting of equal size modules. Furthermore, it cannot handle conflicting preferences. Moreover, the prohibitive computational cost means that the algorithms used in FADES are not very efficient. Above all, it does not engender a diverse set of layout alternatives, a key requisite in generation of LD decision alternatives.

Kumara et al. (1988) present a machine layout design ES (IFLAPS) that deals with the one-to-one assignment type scenarios. It employs a few simple rules of thumb consisting of deterministic steps, which means that it neither affords any actual optimization nor furnishes any diversity in alternatives. IFLAPS requires a significantly high degree of user inputs and interventions and it does not provide functionalities to modify or refine the alternative generated by the system.

Malakooti and Tsurushima (1989) report an ES for multiple-criteria FLD (ES-MCFL) that employs a forward chaining reasoning mechanism. Authors argue that despite the quantitative nature of MCDM, the ability to handle multiple conflicting goals might resemble experts' cognitive treatment of subjective and uncertain preferences. However, ES-MCFL considers only one criterion at a time based on priority rules and does not impart the requisite flexibility and robustness to the system. Furthermore, it uses mostly crisp data, crisp logic, and deterministic heuristics. In order to generate alternatives, users are required to change the priorities and repeat the procedure. Consequently, the solutions do not exhibit diversity. Further, the user interface is not designed to permit decision-makers to manipulate and refine a given alternative. Moreover, the system cannot efficiently handle even modestly large problems.

Heragu and Kusiak (1990) presents a Knowledge-based Machine Layout (KBML) system that tackles one-to-one assignment type scenario. It is claimed to be capable of solving relatively larger problems in comparison to other KBLD systems existing at that time. It employs both quantitative and qualitative data. However, the crisp nature of data means it cannot adequately capture subjective and uncertain dynamics of the problem domain. Furthermore, conflicting preferences require user intervention. KBML employs various models and algorithms, each of which is suitable to some specific scenario, with a hope that a collection of models would cover most of the scenarios. KBML requires manual modification in parameters to generate new feasible solutions and may require several uninformed iterations before producing a workable solution. Furthermore, the deterministic nature of algorithms does not afford an adequate level of optimization and diversity in alternatives. In addition, the computational cost of procuring a viable alternative is still quite prohibitive.

SightPlan is an ES that generates layouts for temporary facilities on construction sites (Tommelein 1997). However, it neither provides ways to incorporate soft constraints and preferences nor it cannot handle conflicting preferences and requires user to manually rectify conflicts. In addition, the layout solutions do not have any diversity, a key requirement in providing design support to LD experts.

12.4.3 Limitations of Existing Knowledge-Based Approaches

Most existing Knowledge-based Layout Design (KBLD) systems are not very robust and flexible, as users might want or as the state of affairs might require.

Such lack of robustness and flexibility are a result of various factors. Here we describe some of the more salient factors.

Scope: In general, a relatively simpler version of the one-to-one assignment type LD scenario is tackled. Such problem formulations have some important applications in various work domains like machine or job shop LD. However, these formulations do not suffice for most LD domains. Consequently, the existing systems do not seem to be effective even in modestly subjective and complex situations.

Scalability: Existing KBLD systems may handle only small-scale problems reasonably fast. However, even for modestly large problems, the time required to solve the problem through these systems could be prohibitive. More general LD scenarios require solutions for large-scale continuous space layout problems consisting of unequal size modules with relatively little computational efforts.

Diversity of Alternatives: In general, heuristics employed for obtaining layout solutions are deterministic in nature. In some KBLD systems, it may involve adding a few production rules to guide the optimization search process. Nevertheless, despite some claims to the contrary, these KBLD systems do not present a diverse set of superior layout alternatives. Nevertheless, the diversity in alternatives is a key ingredient in providing decision support in such complex problem domains.

Quality of Alternatives: The quality of solution alternatives is another core issue in layout decision analysis and design. The deterministic nature of LD algorithms and the lack of diversity in decision alternatives mean that the existing KBLD systems require many reruns before a satisficing layout alternative is obtained. The primary reason is the difficulty in modeling sub-cognitive and implicit preferences as well as difficulty in quantifying the qualitative determinants of layout fitness.

Transparency: The existing KBLD systems offer little or no explanation facilities. Towards this end, simply providing the sequence of the rules employed in reaching a decision may still be considered sufficient. However, relating the accumulated heuristic knowledge to deeper understanding of the domain is still elusive.

Learnability and Reusability: It should be noted that developing an ES for such a complex problem as LD might take efforts equivalent to several scores of person-years (Walenstein 2002). Conceivably, such gigantic and concerted efforts are hard to justify if most system improvements and adaptations call for significant and time-consuming additional labor from its developers (Negnevitsky 2002) Consequently, there is a pressing need for developing ES that learn and update knowledge in an automated manner. Most existing KBLD systems do have an ability to learn from experience and user behavior.

Interactivity: The interactivity in KBLD systems would enable swift change of rules, parameters, algorithms, priorities etc. (Ligget 2000). However, most existing KBLD systems lack user interface that affords effective and interactive analysis and design. Apparently, the LD practitioners themselves designed most interfaces. Thus, these lag considerably in interactivity, usability, and suitability to the work domain.

12.5 Proposed Intelligent Approach to Layout Design

It has been noted that the computer-based layout design algorithms could not replace human judgment and experience, as these algorithms do not always capture the qualitative and intelligence aspects of layout design (Tompkins et al. 2002). Nevertheless, it is often effortless for experts to visually inspect a layout alternative and endorse its acceptability or otherwise. Conceivably, there are strong prospects for devising some incomplete models and intelligent methods to supplement human erudition and intuition. For instance, computerized generations of alternate layouts could provide efficacious support to the layout analyst by aptly addressing some of the complex problem dynamics. Indeed, the possibility of significantly enhancing the productivity of layout analyst and the quality of final solution through automated and expedited production, analysis, and treatment of a large number of superior layout alternatives has been advocated and sought since long (Bazaraa 1975). The popular solution approaches have their strengths and weaknesses. The usual tradeoff involved between the flexibility in incorporating the problem-specific details and the exhaustiveness of the search in various LD optimization tools is depicted in Fig. 12.2 [17].

In Fig. 12.2, on one end are enumerative search techniques, which are superior in terms of exhaustiveness of solution space search. However, such general techniques incorporate very meager amount of problem-specific information and their application is marred by the process speed and computational

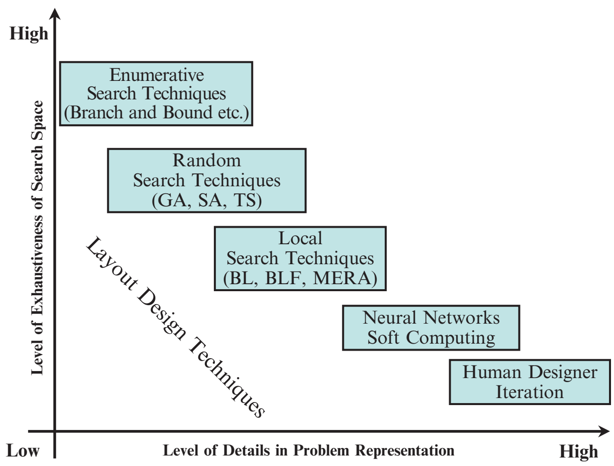


Fig. 12.2. Flexibility and robustness of various layout design approaches (Ahmad 2005, Chung 1999)

complexity. On the other end, human designers command high level of flexibility and the capability of incorporating detailed problem-specific information into the design process. However, the cognitive and information processing limitations of human designers translate into inadequate of search in the solution space. Between these two extremes are techniques that provide various degrees of flexibility through selection of tools, algorithms, and parameters that incorporate varying level of details in the representation of problem-specific information and design process. Conceivably, an intelligently formulated hybrid approach involving metaheuristics (random search), placement algorithms (local search), soft computing modeling and computational tools (approximate reasoning), and human intuition could deliver a higher degree of flexibility and efficacy.

In short, various modeling and computational tools and heuristics could help in characterizing possible outcomes, and the behavioral data may express some salient points about the designers' behavior and preferences (Ahmad et al. 2004). In this regard, computerized decision support tools may be viewed as a mechanism for *redistribution of cognition* (Welgama et al. 1995). Such tools provide support through various means such as *process distribution, data distribution, plan distribution*, etc. (Walenstein 2002).

Our research framework is based on the Expert System (ES) paradigm for facilitating intelligent decision support in layout design. The emphasis of this research is not on the pursuit of some perfect system but rather on the development of a tool that could supplement the knowledge, experience, and design intuition and other cognitive resources of human layout designer. Our selection of ES as a research paradigm is inspired by such inherent characteristics of an ES as the encoded knowledge, the separation of domain knowledge from the control knowledge, the ability to reason under uncertainty, the explanation facility, the knowledge acquisition capability, and the interactive user interface. A traditional ES paradigm is shown in Fig. 12.2. However, an efficient and effective means of tackling the subjectivity and uncertainty in the layout design problem requires complementing the traditional ES paradigm through various intelligent components. Such intelligent components would afford effective, efficient, and robust means of capturing and utilizing subjective and uncertain design preferences, while generating a diverse suite of superior layout alternatives. Consequently, our research paradigm, as depicted in Fig. 12.3, contains some components that are not associated with traditional expert systems. These include an Intelligent Layout Generator (ILG), a Preference Inferencing Agent (PIA), and a Preference Discovery Agent (PDA). It should be noted that this research framework evolved during the course of this research as more insights are gained about the structure of the problem at hand and the underlying dynamics.

As mentioned, an array of efficient algorithms for generating superior and diverse layout alternatives is an important step in automating the layout design process. We use a hybrid fuzzy-genetic Intelligent Layout Generator (ILG) towards this end. The intelligence aspect emerges from the

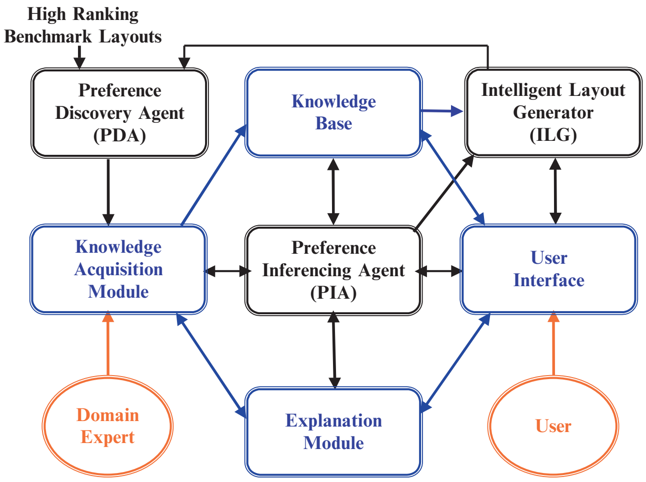


Fig. 12.3. Intelligent System for Decision Support/Expert Analysis in Layout Design

employment of fuzzy rules/preferences in obtaining penalties and rewards for some composite genetic fitness evaluation function. Accordingly, a fuzzy Preference Inferencing Agent (PIA) seems to be a rational component for such a decision-aid tool.

As noted, the layout design rules and preferences are both implicit and dynamic in nature. People learn new concepts and outgrow old ideas, thus pronouncing the necessity for re-learning of design rules by layout designers. Such an implicit and evolutionary character of preferences suggests that an online Artificial Neural Network based Preference Discovery and Validation Agent (PDA) could augment the overall power of the system by discovering some pattern of design rules and preferences in an automated and self-updated manner.

It should be mentioned that not all details of these components are made explicit in this framework for parsimony sake. For instance, our PDA is designed in a manner that it could furnish the learned knowledge in the form of usable knowledge by creating preference profiles of decision makers. As such, PDA would not require any explicit and separate knowledge acquisition module. Here we provide further details of various components of IdeaL, including their philosophy and operation.

12.5.1 Intelligent Layout Generator

We present a Genetic Algorithms (GA) based approach for building an Intelligent Layout Generator (ILG) by employing various layout design heuristics, including some new, fast, and efficacious ones. The intelligence aspect comes from the employment of penalties/rewards or preference weights, furnished by a Preference Inferencing Agent, in the evaluation of a genetic fitness function.

The primary task involved in automating the LD process is to produce superior layout alternatives for further consideration and treatment by decision makers (Akoumianakis et al. 2000, Tompkins et al. 2002). In this regard, past studies have demonstrated that Genetic Algorithms provide a promising search and optimization approach (Abdinnour-Helm and Hadley 2000, Ahmad et al. 2006, Youssef et al. 2003). Our system incorporates experts' knowledge and user preferences in the LD process through composite fitness functions of the ILG. This fitness function utilizes crisp preference weights furnished by the Preference Inferencing Agent.

It should be noted that we carried out preliminary experiments with various layout design problem formulation including QAP, QSC, and 2D-BPP. Furthermore, we employed several popular solution approaches including analytical and heuristic solution methodologies as well as such metaheuristics based search mechanisms as GA, SA, TS, NE, and RS, etc. Our preliminary studies resulted in the selection of 2D-BPP as the formulation for this research due to its more generic and natural characterization of the layout design problem. In addition, we adopted GA, in conjunction with some efficient placement heuristics, as a solution methodology due to its global scope and non-deterministic search mechanism as well as potential to furnish a diverse set of superior layout alternatives.

In short, these preliminary studies were the driving force in the selection of the approach we employed in this research. It involves hybridization of the global search mechanism through GA and the local optimization through deterministic placement heuristics. Indeed, our approach has some innate characteristics, discussed later on, which are advantageous in providing effective decision support in layout design.

Most of the existing research applies GA in solving layout problem involving identical modules to be placed at identical locations. Such a problem can be treated as a relatively simpler one-to-one assignment of identical modules to the given cells/locations. In relatively advanced scenarios, the size of modules is considered fixed while leaving the determination of the shape of module to the solution procedure. Still, some advanced research work employs GA in solving problems involving oriented modules with fixed dimensions, which are to be placed in a two-dimensional plane. However, employing GA in such more advanced and generic layout design scenarios requires efficient and efficacious decoding or placement heuristics. Such heuristics are important in order to generate layout alternatives in a timely fashion. Indeed, the importance of such pre-processor algorithms in terms of efficiency, efficacy, and reliability cannot be overemphasized. Various decoding or placement heuristics are available in the literature, for instance, BL (Dowsland et al. 2002, Jakobs 1996), IBL (Liu and Teng 1999), BLF (Chazelle 1983), and DP (Leung et al. 2003). However, there is a relative dearth of decoding algorithms that are not only fast but also robust and effective in furnishing superior layout alternatives with higher aesthetic contents. In order to address this shortcoming, we have proposed some very effective decoding or placement heuristics.

Details of these algorithms as well as our vision and implementation of ILG are provided here.

12.5.2 Fitness Evaluation Metrics

As already noted, the LD problem involves such a plethora of subjective and uncertain considerations that no single objective could solely be used to generate layout alternatives. However, automated LD systems require some fitness quantification and evaluation mechanism in order to guide the search to superior solutions. We, therefore, propose the use of some hybrid fuzzy-genetic fitness function that would combine multiple objectives arising from various layout design considerations. As such, various determinants of the layout utility are combined through some crisp weights or Significance Parameters (SP) to penalize deviation from the desired values or Preference Parameters (PP). These significance and preference parameters may be determined by the layout planners or through the PIA using the existing knowledge. As a preliminary research model, we envisaged the following major categories of design preferences as determinants of layout fitness: Intrinsic Utility of a module, Inter-Module Interaction, Space Utilization, and Qualitative Fitness or Aesthetic Appeal. Intrinsic utility of a module is the utility a module brings when it is included in a layout design. For simplicity sake and without any loss of generalization, we ignore inherent utility of a module in our discussions.

We consider inter-module interaction as an important determinant of layout fitness. IdeaL has been equipped with functionalities for modifying these inter-module interactions in an *interaction matrix*, containing the interaction between all pairs of modules. An element of this matrix is denoted by $f_{i,j}$ and represents the flow between any two modules M_i and M_j. We calculate it as the sum of mutual distances between geometric centers of all pairs of modules or the total Inter-Module Distances (*IMD*).

The space utilization is among the more popular layout design fitness metrics and the literature proposes the Contiguous Remainder (*CR*) or the 'reusable trim loss' as a more appropriate measure of space utilization (Jakobs 1996, Liu and Teng 1999). The *CR* refers to the largest contiguous vacant portion of the packing space available for further placements (Ahmad 2005, Jakobs 1996, Liu and Teng 1999). In other words, *CR* is the empty area on a bin outside the edges of the boundaries created by the packed modules in a layout, as shown in Fig. 12.4. Conceivably, a larger value of *CR* implies that more space is available for further placements.

The Contiguous Remainder can be calculated by using the following expressions:

$$CR = Page\ Area - Total\ Module\ Area - Trapped\ Dead\ Space \tag{12.1}$$

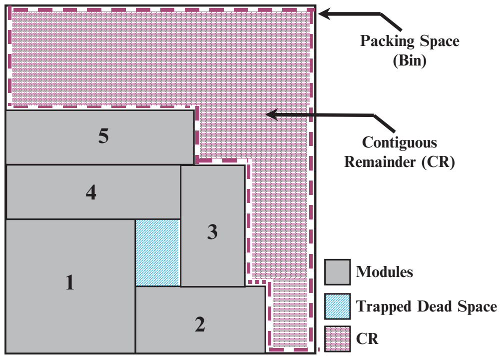


Fig. 12.4. Elaboration of the concept of Contiguous Remainder

If H and W are the height and width of the packing space and h_i and w_i are the height and width of an module M_i, then:

$$CR = H \times W - \sum_{i=1}^{n} w_i h_i - \textit{Trapped Dead Space} \qquad (12.2)$$

A dual of *CR* is the White Space Level (*WSL*), which is a normalized function and suits the GA and MCDM paradigm more than the *CR* and calculated as follows:

$$C\widehat{R} = WSL = \frac{CR}{\sum_{i=1}^{n} w_i h_i} \times 100 \qquad (12.3)$$

The Trapped Dead Space is an important measure of space utilization in itself as well as in calculation of other metrics as *CR* and *WSL*. Its calculation however is not straightforward. An algorithm was developed for IdeaL since no algorithm for the exact calculation of the trapped dead space or the contiguous remainder was found in the published literature. IdeaL calculates the exact dead space by detecting the trapped spaces through a digital scanning of the packing created at any instance when a module is placed. This algorithm keeps track of all areas occupied by the placed modules and thus finds the trapped dead spaces as the areas not occupied by any module. Despite all the subjectivity and uncertainty involved in calculating the intrinsic utility of a module, the inter-module interaction, and the space utilization, we classify these as quantitative measures of layout fitness. The rationale is that these measures may be quantitatively captured in an automated or semi-automated fashion with relative ease, given that the required data is complete and known with certainty.

Aesthetic values are subjective measures of layout quality. Such values cannot easily be defined in specific terms and usually depend on users' personal judgments. Different people may rate the perceived aesthetic appeal of a given layout differently. Consequently, we classify aesthetic appeal of a layout as a qualitative measure of fitness. It should be noted that GA are also known to be promising search strategy when fitness functions involve qualitative decision variables [10]. However, to the best of our knowledge, no earlier study has compared computerized layout design algorithms in terms of ability to generate solutions with higher aesthetic appeal.

12.5.3 Genetic Algorithms based Optimization

The Genetic Algorithms (GA) based approach for solving the layout design problem requires determining several critical features including an adequate encoding scheme, an adequate population size, an adequate set of genetic operators, an adequate fitness function, an efficient module placement strategy, and adequate stopping criteria. It should be noted that final set of evolution operators (selection, crossover, mutation, and replacement) and parameters (population size, crossover rates, mutation rates, and termination criteria) would be determined after extensive experimentations with the GA. Nevertheless, it has been argued that the effectiveness of GA methodology is largely insensitive to the exact values of these parameters (Tate and Smith 1995).

The GA encoding scheme for the layout design is a sequence of modules similar to the one adopted by Tate and Smith (Tate and Smith 1995). The sequence S of the module indices (or names). For example: *Sequence of Modules* $= S = \{12,\ 4,\ 9,\ 25,\ 11,\ 47,\ 2,\ 8,\ 16,\ 13,\ 31,\ 45,\ 29,\ 19,\ 33,\ 5,\ 19,\ 7,\ 34,\ 50\}$. This example shows how a sequence of 20 modules, out of a set of 50, to be placed in a given bin. The total length l of the sequence S could be specified either by the expert or possibly be determined by using the maximum number of modules that could be placed on a single bin, amount of white space desired, etc.

We used a pre-specified and static population size P in each generation in evolution process. The initialization step in the GA randomly generates P sequences of modules $(S_1,\ S_2, \ldots S_P)$. Previous studies have shown that a population size of 10–20 provides superior results (Tate and Smith 1995, Jakobs 1996).

In GA, genetic evolution of population creates new layout solutions through genetic operators (crossover and mutation of individual layouts from previous generation). The means of performing these operations must be defined for the layout design problem. A variety of genetic operators could be suggested for the GA. However, we limit ourselves to genetic operators used by Tate and Smith (1995) and Jakobs (1996) for solving layout design problems. These constitute the most popular although only a small extract of possible operators.

The selection operator selects individual layout solutions for genetic operations. We used the rank based selection strategy commonly known as Roulette Wheel selection, one of the most commonly used selection strategies, which is biased towards selecting the fitter solutions for further evolution (Negnevitsky 2002).

In mutation, mutating a single solution generates new individuals. In the context of layout design problem, mutation results in small changes in an existing layout. The mutation rate is selected to be high (around 50%). The reason is that any given chromosome contains only a small subset of the given modules and high mutation rate would ensure that higher chances of incorporating all or most of the modules in test solutions. Furthermore, higher cost of placement algorithms pronounces the need of using 'incremental' GA. Consequently, a higher mutation rate ensures diversity in the population of layouts (Ahmad 2005, Jakobs 1996). The following mutation operators are used in the ILG:

1. Tate and Smith (1995) proposed following set of mutation operator: Reverse the subsequence of the sequence in the mutating layout solution (random selection of the mutating solution and mutating subsequence).
2. Jakobs 1996 used the following set of mutation operator: Exchange elements of two randomly selected layout subsequences.
3. Replace a randomly selected module with a randomly selected module.

During crossover, one or more offspring layouts are derived from two or more parent layouts. In the context of layout design problem, crossover results in combining parts of two existing layouts in order to generate a new layout. The following crossover operators will be used on two parents (say S_j and S_k) selected randomly based on their ranks in the population. Previous studies have demonstrated the success of these operators (Tate and Smith 1995, Jakobs 1996).

1. Tate and Smith (1995) Crossover consists of following steps:
   a. Fill each position in the offspring layout by randomly selecting a gene present at the same position from the first or second parent layout (resolving conflicts).
   b. Insert leftover genes in order (or in random order) to fill in the blanks (unresolved conflicts).
2. Jakobs (1996) Crossover consists of following steps:
   a. Copy q elements of the sequence S_j at a random position p in the new sequence S_{new}. It should be noted that $1 \leq p,\ q \leq n$.
   b. Fill up the remaining elements of S_{new} with other elements of S_k.
3. Append a Randomly selected subsequence from one parent to another.

Traditional GA generates P offspring layouts before sorting out the poor ones by selection. We argue that module placement strategies are computationally very costly. Consequently, we propose that GA sort out the worst individual after a new offspring layout is created, regardless of the fitness of the offspring, on an ongoing basis. As a result, 'superior' offspring could influence the layout solution quality. However, such strategy pronounces the need for high mutation rate to ensure population diversity. An approach similar to his one has proved to be effective and superior for the layout design problem in (Jakobs 1996). This strategy results in a 'steady state' or 'incremental' GA as opposed to a 'generational' GA where multiple offspring are created to replace the current population.

The most taxing and application specific task in any particular problem domain exploiting GA is definition of the fitness function. The fitness function is used to differentiate between a 'good' and a 'poor' layout solution. A fitness function should be a well-thought function, as the GA will converge on layout solutions deemed 'fit' by this fitness function. As discussed, a layout design problem involves such a plethora of considerations that no single objective could solely be used to generate alternate layouts. We, therefore, propose a genetic fitness function that combines multiple objectives in terms of rewards/penalties arising from various layout design considerations. The various determinants of layout utility are combined through some crisp weights or preference parameters.

We terminated the GA when the improvement in the fitness of new population over the preceding population is less than a certain value (say *0.1%* or so) or after a certain number of Generations. However, the user would finalize this criterion after performing some focused experimentations with GA.

12.5.4 Proposed Decoding Heuristic

As discussed in Sect. 12.3, existing decoding algorithms lack the requisite efficiency and efficacy. Such shortcomings are more pronounced when layout evaluation criteria include such aesthetic values. In this section, we outline a new, efficient, efficacious, and robust placement algorithms developed for constructing the actual layouts with higher aesthetic contents [3]. The placement algorithm works with an ordering of modules obtained through some non-deterministic and evolutionary metaheuristic-based approach, which is GA in case of IDEAL. The new module placement algorithm is inspired by the fact that for any given packing space the number of modules at hand for placement is a small integer. Moreover, if we confine our placement possibilities only to the corners of 'in-place' modules then for a particular module there exist at most $O(n)$ possible locations. Accordingly, the combinatorial complexity should not pose a significant problem if some intelligent and fast pseudo-exhaustive exploration is carried out in a hierarchical manner for enhancing the space utilization and the layout quality. The primary motivation in our quest for improved heuristics was our desire to generate layouts with both higher aesthetic contents and better space utilization. Consequently, we were willing to make a tradeoff in speed in order to get improved quality. Nevertheless, comparative studies have shown that the proposed algorithm is more efficient in the metaheuristic-based layout optimization than other existing heuristics.

We call the proposed placement algorithm as Minimization of Enclosing Rectangle Area (MERA). The name is inspired by the underlying notion where a reduction of the rectangular area of the packing pattern, called Area of Enclosing Rectangle or *AER*, is sought during all placement decisions with a bias term favoring lower placements. The optimization part in the placement strategy is not an extensive or expensive optimization but a sort of a heuristic refinement – a pseudo-exhaustive search. Such a hierarchical optimization scheme facilitates improvement in space utilization as well as quality of layouts. It should be noted that IDEAL also contains several intelligent adaptations of MERA to provide greater flexibility and power to the user.

The algorithm (Ahmad et al. 2006) proceeds by investigating the placement prospects for all four corners of an in-coming module at all four corners of all in-place modules seeking to find the minimum value of the composite objective function that includes a bias in favor of placement at the bottom-left position in the layout, which is a general packing preference in various placement heuristics or LD contexts such as bin-packing.

In MERA, each in-coming module can be placed at a maximum of $16(i-1)$ corner points (a very weak upper bound) where $i-1$ modules are previously in place. As such, theoretically the MERA algorithm also has the same $O(n^\wedge 2)$ cost as for BL and IBL (Jakobs 1996, Liu and Teng 1999). Moreover, some increase in the computational complexity is considered quite rational if significant improvements in terms of both quantitative and qualitative fitness metrics are realized, as demonstrated by the comparative analyses.

12.5.5 Comparative Evaluation of Decoding Heuristics

In order to test and validate the efficiency, efficacy, and robustness of our placement algorithms in producing layout of higher aesthetic contents, we employed both automated capturing of quantitative measures as well as visual evaluations by experts in layout design. We employed some randomly generated and some benchmark problems from the literature for our studies.

A computer program was written in Visual BASIC to implement the BL, IBL, BLF, MERA, and the GA based optimization component including various fitness evaluation functions. The computer program is used for comparative analyses on Intel Xeon 3.06 GHz processor with 256 MB RAM under Windows XP.

Apart from quantitative analyses based on contiguous remainder and inter-module distances, three facility layout design researchers and practitioners were asked to provide subjective rating of some layout alternatives in terms of symmetry. These experts have decades long experience in teaching, researching, and practicing in layout design applications. These experts had no knowledge of the algorithm/method used for generating these alternatives. Furthermore, they did not have any indication of fitness metrics/values used by us. In addition, these experts were under no time constraint for furnishing their ratings. All three experts have decades long experience in teaching, researching, and practicing in layout design applications. These ratings were on a scale of 1–10 with a higher score representing higher aesthetic value perceived by the expert. We want to emphasize that a layout quality rating of 10 represents a highly symmetric layout configuration, which usually cannot be achieved for problems consisting of randomly generated unequal modules or when modules dimensions have high variability. Consequently, we found that a Layout Quality rating of around 5 implies that the layout alternative is quite superior for the given problem.

We used several benchmark problems from the literature for our comparative studies. We initially employed a Random Search approach for our comparative studies by generating 100 random sequences of modules. As already mentioned, Random Search and Naive Evolution are among the most effective search strategies, though not at par with GA or SA, for layout design problems. The relative performance of the BL, IBL, BLF, and MERA placement strategies for 100 random sequences of each benchmark problem instance is discussed in (Ahmad et al. 2006). Results have shown that MERA outperforms the existing algorithms by wide margins. The proposed algorithm generate superior outcomes in terms of the Contiguous Remainder *CR* (the higher the better), the Inter-Module Distances or *IMD* (the lower the better) and the layout Quality Rating *QR* (the higher the better). The performance gains are more pronounced for larger problems. This superior performance can be shown as statistically significant using means and standard deviations.

We also employed GA based metaheuristic search in our comparison. The average of ten GA runs for the 100-module problem with a population size of 50, a mutation rate of 0.8, and a crossover rate of 0.2 is shown in Table 12.1. It can be seen that MERA outperforms the existing algorithms by wide margins.

Table 12.1. Comparison of Decoding Heuristics with GA search

Objective	Tech.	Best Fitness (% difference from optimal)
CR	**BL** + GA	3432 (−31.4%)
(Optimal = 5,000)	**IBL** + GA	3905 (−21.9%)
The Higher the Better	**BLF** + GA	4235 (−11.3%)
	MERA + GA	**4709** (**−5.8**%)
IMD	**BL** + GA	553459.5 (+1.7%)
(Reference = 536,000)	**IBL** + GA	521419.6 (+7.4%)
The Lower the Better	**BLF** + GA	483010.3 (+14.2%)
	MERA + GA	**450759.9** (+**19.9**%)
QR	**BL** + GA	1.5
(Ideal = 10)	**IBL** + GA	1.75
The Higher the Better	**BLF** + GA	3.5
	MERA + GA	**5.25**

12.5.6 Fuzzy Preference Inferencing Agent

Here we provide details about modeling of, and inferencing from, subjective and uncertain preferences as well as the design, implementation, and working of the Preference Inferencing Agent.

The brain of any ES is an Inference Engine that contains general algorithms capable of manipulating, and reasoning about, the knowledge stored in the knowledge base for solving problems by devising conclusions (Turban and Aronson 2001). The inference engine in an ES is kept separate from the domain knowledge and is largely domain-independent.

A major problem in building intelligent systems is the extraction of knowledge from human experts who think in an imprecise or fuzzy manner. The same is true with the layout design problem where the knowledge associated with the layout decision analysis and design is usually imprecise, incomplete, inconsistent and uncertain. In the scope of our research, the term *imprecision* refers to values that cannot be measured accurately or are vaguely defined. Likewise, *incompleteness* implies the unavailability of some or all of the values of an attribute, *inconsistency* signifies the difference or even conflict in the knowledge elicited from experts, and *uncertainty* suggests the subjectivity involved in estimating the value or validity of a fact or rule.

The inherently vague, differing, and conflicting nature of most LD guidelines and rules renders fuzzy technology an excellent candidate for modeling the system dynamics as well as implementation of the inference engine. Indeed, FL provides a means to work with these imprecise terms and has been successfully employed for automated reasoning in expert systems in various subjective and uncertain work-domains. However, little effort has been done in formalizing such an application of fuzzy logic in LD systems. Nevertheless, an FL based Preference Inferencing Agent seems to be an important component in any LD decision aid tool (Ahmad 2002, 2005, Karray et al. 2000, Raoot and Rakshit 1993).

As such, the underlying concept in IdeaL's inferencing uses a Preference Inferencing Agent (PIA) comprising of fuzzy sets, rules and preferences for obtaining penalties and rewards in the layout fitness evaluation function for ranking and comparison purposes as well as for the automatic generation of layouts. The potential for utilizing FL arises from the fact that it provides a very natural representation of human conceptualization and partial matching. Indeed, the human decision-making process inherently relies on common sense as well as the use of vague and ambiguous terms. FL provides means for working with such ambiguous and uncertain terms (Negnevitsky 2002). Consequently, an FL based PIA is expected to deliver much of the flexibility in the automated LD process that the LD practitioners have always longed for. As such, we deem PIA as one of the core components, along with ILG, in tackling and automating the LD process as well as in furthering the research in this important area. Further details of our vision and realization of the PIA are given in Sect. 12.5.1.

The core concept involves employing a PIA comprising of fuzzy sets, rules, and preferences in obtaining penalties and rewards for the hybrid fitness evaluation functions as well as various critical parameters for ILG and PDA. The primary benefit of fuzzy rule-based system is that its functioning mimic more of human expert rules. The traditional rigid and myopic fitness functions do not serve well in such complex, subjective, and uncertain problem domains as layout design. Indeed, multi-criteria fitness functions are deemed more appropriate for automatic generation, evaluation, and comparison of layout alternatives. However, IDEAL has provisions for decision-maker to specify Significance Parameter (SP) and Preference Parameter (PP) in both crisp and fuzzy manner, thereby increasing the flexibility and the ease with which decision-makers may creatively adapt their preferences.

Fuzzy-Normalized Weighted Sum Loss Function

Here we propose a novel approach to f-MCDM for multi-dimensional multi-attribute decision problems, in general, and layout decision analysis, in particular. Our approach draws from the relative simplicity of FWSM and efficacy of relative fitness values (as in AHP). It is inspired by Taguchi's quality loss function where any deviation from the nominal values results in a reduction in utility. Accordingly, our approach involves employing the normalized values of principal layout fitness metrics and calculating the deviation from some preferred nominal values. This deviation, in turn, is used to calculate penalties based on the weight or significance S_κ assigned to each fitness attribute κ. We term this approach as Fuzzy Normalized Weight-Sum Loss Function (f-NWSLF).

Conceivably, the selection of these benchmarks for normalization in such subjective and uncertain work domain as layout design remains a contentious issue and constitutes an open research question. As such, the benchmarks employed for normalizing each fitness dimension may be contended. However, the selection of these benchmarks was made after extensive preliminary studies with a range of intuitively selected benchmarks, which revealed these as satisficing benchmarks for our purposes.

In essence, the penalty function calculates the weighted sum of penalties, where weights are the significance S_κ assigned to a fitness attribute κ and penalty is the deviation of normalized fitness value $\hat{f}_\kappa$ from its preferred value P_κ. In this manner, we are combining the powers of three effective MCDM techniques. This penalty function may be made more or less precipice using a parameter $\psi > 1$. A value of $\psi > 1$ would result in a more precipice loss function, whereas a value of $\psi < 1$ would result in relatively flat loss function. It should be noted that if ψ is not a multiple of two then it requires the penalty function to be absolute deviation from $\hat{f}_\kappa$. However, currently we are using the penalty as proportional to the square of deviation (i.e. $\psi = 2$), as follows:

$$F_{f-NWSL} = \sum_{\kappa=1}^{p} S_\kappa \left\{ \left| \hat{f}_\kappa - P_\kappa \right|\right| \right\}^{\psi}$$

It should be noted that certain parameters could have significant interaction with one another affecting more than one value of crisp weights used subsequently in the layout evaluation phase. In addition, the question of developing more effective and robust layout fitness metrics remains open for further research in MCDM field.

Working of Preference Inferencing Agent

In order to elaborate the working of the PIA, we consider a scenario where the small size of the packing space would not permit placement of all the given modules in the layout configuration, a common scenario in practice. We consider the same 100-module problem used in Sect. 12.5.1, but the reduced size of the packing space precludes the placement of all 100 modules.

In our example, the amount of 'white space' and the 'size of bin' affect the maximum number of 'bin modules' that could be placed in a single bin or packing space. This important parameter determines the efficiency and efficacy of the whole process. For instance, it would affect the length of chromosome chosen for a GA used in the ILG, determining the search space, dramatically affecting the efficiency and quality of results. IT is because employing a chromosome size of 100 would result in unnecessary search and slow progression of the GA based optimization process.

In our example, we let *x, y*, and *z* (*white_space, bin_size*, and *chromosome_size* respectively) be the linguistic variables; *A1, A2*, and *A3* (*small, medium*, and *large*) be the linguistic values determined by fuzzy sets on the universe of discourse *X* (*white_space*); *B1, B2, B3* and *B4* (*small, medium, large* and *ex-large*) be the linguistic values determined by fuzzy sets on the universe of discourse *Y* (*bin_size*); *C1, C2*, and *C3* (*small, medium*, and *large*) be the linguistic values determined by fuzzy sets on the universe of discourse *Z* (*chromosome_size*). The membership functions for these linguistic variables are shown in Fig. 12.7. The complete set of fuzzy rules for determining *choromosome_size* using *white_space* and *bin_size* is provided in Table 12.2. Our example consists of a simple two-input and one-output scenario with the following two fuzzy rules specified by an expert:

We used the Mamdani-style inference method, as it is the most popular technique for capturing experts' knowledge, (Negnevitsky 2002) Using this technique, the crisp value for the chromosome size came out to be 27 (Ahmad 2005).

In order to evaluate the effect of the *chromosome_size* as determined by the PIA, we ran 1,000 iterations of the GA with chromosome sizes of 27 and 100 employing MERA as the decoding heuristic. The average time per GA iteration with a chromosome size of 100 was 15.43 s. In contrast, the average

Table 12.2. Fuzzy rules for determining the chromosome size

Rule 1:	**Rule 2:**
If x is $A2$ (*white_space* is *medium*) Or y is $B3$ (*bin_size* is *large*) Then z is $C2$ (*chromosome_size* is *medium*)	If x is $A3$ (*white_space* is *large*) Or y is $B4$ (*bin_size* is *ex-large*) Then z is $C3$ (*chromosome_size* is *large*)

		Bin Size			
		Small (B1)	Medium (B2)	Large (B3)	Ex-Large (B4)
White Space	Small (A1)	*Small*	*Small*	*Medium*	*Medium*
	Medium (A2)	*Small*	*Medium*	*Medium*	*Large*
	Large (A3)	*Medium*	*Medium*	*Large*	*Large*

time per GA iteration with a chromosome size of 27 was only 0.316 s. It elaborates how a simple adaptation of a GA parameter through fuzzy rules and inferencing could affect the efficiency of the overall process. Furthermore, this example illustrates how vague linguistic rules can be used to derive important and useful crisp values. Likewise, the PIA can be used to furnish other parameters for subsequent use. Our preliminary studies show that fuzzy logic constitutes an effective inferencing tool in LD, providing greater flexibility, expressive power, and ability to model vague preferences.

12.5.7 Preference Discovery and Validation Agent

The reliability and effectiveness of PIA significantly depends on the reliability of preferences. The task of extracting knowledge from experts is extremely tedious, expensive, and time consuming. In this regard, the implicit and dynamic nature of preferences as well as efforts required for building and updating an expert system underscore the need for automated learning. Indeed, learning is an important constituent of any intelligent system (Negnevitsky 2002). However, a traditional ES cannot automatically learn preferences or improve through experience. Here we describe a small-scale Preference Discovery Agent (PDA) for testing the idea of automated preference discovery and revision in LD.

An automated learning mechanism could improve the speed and quality of knowledge acquisition as well as effectiveness and robustness of ES. Incidentally, Artificial Neural Networks (ANN) have been proposed as a leading methodology for such data mining applications. ANN can especially be useful in dealing with the vast amount of intangible information usually generated in subjective and uncertain environments. The ability of ANN to learn from historical cases or decision-makers' interaction with layout alternatives could automatically furnish some domain knowledge and design rules, thus eluding tedious and expensive processes of knowledge acquisition, validation and revision. Consequently, the integration of ANN with ES could enable the system to solve tasks that are not amenable to solution by traditional approaches (Negnevitsky 2002).

Fortunately, the layout design problem renders itself to automatic learning of non-quantifiable and dynamic design rules from both superior layout designs and test cases. Furthermore, it is possible to automatically learn some decision-makers' preferences from their evaluation and manipulation of accepted or highly ranked layouts using some online ANN based validation agent. However, in the absence of fully functional core components like ILG and PIA, which would exploit the layout design preferences, an effective PDA could not be developed and tested. Consequently, we have given PDA a lower priority in developing IdeaL. Nevertheless, here we provide design and implementation of a small-scale prototype of PDA for demonstrating the viability of concept. In future, we intend enhance capabilities of our PDA and to employ Reinforcement Learning technology to complement ANN through incremental learning.

In order to test our concept, we used well-known Multi-Layer Perceptron Network (MLP). We employed a Feed Forward Multi-Perceptron ANN as we were able to generate a modest number of instances for training and testing reported in (Ahmad 2005). In our PDA, we used a fully connected artificial neural network with one hidden layer. The network consists of two input neurons, three hidden neurons, and a single output neuron forming a directed acyclic graph. The inputs to PDA consist of Module Tightness (X_1) and Symmetry of Distribution (X_2), the later one is a subjective measure of fitness and details of which can be found in (Ahmad 2005, Mak et al. 1998). Furthermore, the output of the PDA is the rating of the layout (Y) for the given inputs. The number of hidden nodes in a network is critical to the network performance. A neural network with too few hidden nodes can lead to underfitting and may not be able to learn a complex task, while a neural network with too many hidden nodes may cause oscillation, overlearning/memorization, and hamper the ability for generalization (Ahmad 2005, Negnevitsky 2002). The decision on the architecture of an ANN is typically done through a trial-and-error. We found a hidden layer with three neurons sufficient for our purposes.

We used MATLAB to code our algorithm for training the PDA based on the popular back-propagation supervised learning paradigm. In this paradigm, the network can be trained by measurement data from the training set. It propagates the errors backwards by allocating them to each neuron in accordance to the amount of this error for which the neuron is responsible. The prediction capability of the trained network can be tested for some test data. The caveat in using the back-propagation algorithm and the MLP is that these require a large number of training examples.

We employed the popular Mean Square Error (MSE) as a measure of performance or convergence. We used a learning rate of 0.01 and programmed to terminate the training of the network after 50,000 epochs or when Absolute MSE goes below 0.001, whichever occurs first. We generated a random permutation of training data set before proceeding to the training of the PDA.

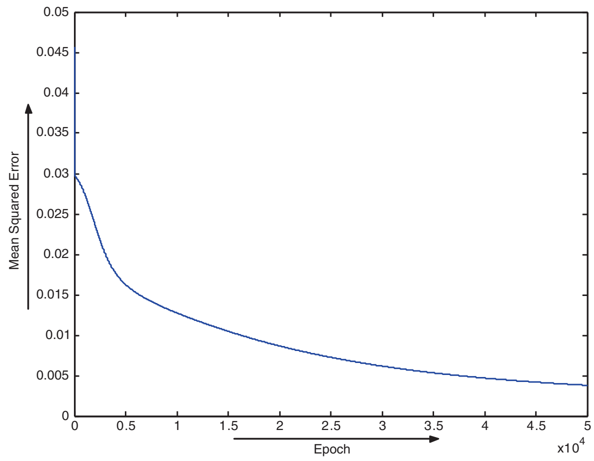


Fig. 12.5. Convergence of the training phase of the PDA

Furthermore, we scaled PDA inputs (X_1 and X_2) and target values (T) in the [0,1] range. As such, the PDA outputs (Y) are also obtained as scaled values in the [0,1] range. The convergence of PDA's training is shown in Fig. 12.5, demonstrating a sound convergence capability of the PDA. For comparison purposes, the Pattern Error, or the difference between the target value and the actual output for the training set of PDA, was less than 4%, indicating the capability of PDA to learn and generalize from the given training instances.

12.5.8 Knowledge Base

Knowledge is the primary raw material in an ES (Walenstein 2002). The conceptual model of the elicited knowledge is converted to a format suitable for computer manipulation through a process called the Knowledge Representation (Negnevitsky 2002). The processes of knowledge elicitation and representation are not necessarily sequential. Typically, knowledge elicitation continues throughout the lifecycle of the system development and its usage as knowledge may be incomplete, inaccurate, and evolutionary in nature.

The knowledge of IdeaL consists of facts and heuristics or algorithms. It also contains the relevant domain specific and control knowledge essential for comprehending, formulating, and solving problems. There are various ways of storing and retrieving preferences/rules including 'If-Then' production rules. Representing knowledge in the form of such traditional production rules enhances the modularity of the system and prompted us to adopt this approach. However, conventional logic based representation does not allow simple addition of new decision rules to the ES without any mechanism for resolving conflicts, thus resulting in inflexibilities that are not conducive to automated LD systems. This furnished another reason for our choice of fuzzy logic modeling preferences and building the inference engine for IdeaL.

12.5.9 Knowledge Acquisition Module

Knowledge acquisition is the accumulation, transmission, and transformation of problem solving expertise from experts or knowledge repositories to a computer program for the creation and expansion of the knowledge base (Turban and Aronson 2001). It should be noted that knowledge acquisition is a major bottleneck in the development of an ES (Jackson 1999). It is primarily due to mental activities happening at the sub-cognitive level that are difficult to verbalize, capture, or even become cognizant of, while employing the usual cognitive approach of knowledge acquisition from experts (Negnevitsky 2002). Consequently, the task of extracting knowledge from an expert is extremely tedious and time consuming. It is estimated that knowledge elicitation through interviews generate between two and five usable rules per day (Jackson 1999).

Knowledge could be derived from domain experts, the existing knowledge, as well as through some automated machine learning mechanism. We intend to formulate our PDA in a manner that could provide knowledge about user preferences in a form readily usable by ILG and PIA. However, the automated knowledge acquisition has not been tackled rigorously in this research.

12.5.10 Explanation Facility

The ability to trace responsibility for conclusions to their sources is crucial to transfer of expertise, problem solving, and acceptance of proposed solutions (Turban and Aronson 2001). The explanation unit could trace such responsibility and explain the behavior of the ES by interactively answering questions. For instance, an explanation facility enables a user to determine why a piece of information is needed or how conclusions are obtained.

Explanation Facilities are vital from both system development and marketing perspectives. These facilitate both debugging of the knowledge base as well as user acceptance and adoption. Such facilities may include user input help facility, design process information, and interrogation facilities. In its simplest form, an explanation facility could furnish the sequence of rules that were fired in reaching a certain decision. Indeed, the capability of an expert system to explain the reasoning behind its recommendations is one of the main reasons in choosing this paradigm over other intelligent approaches for the implementation of our concept.

Once again, a well-designed, interactive, and effective user interface is an important ingredient in enabling a good explanation facility. In addition, incorporation of effective explanation capabilities is elusive without conducting a meticulously designed empirical study with actual users. However, such an extensive study is beyond the scope of this research. However, IdeaL contains a basic explanation capability through which experts can trace the sequence of rules that are used in arriving at certain conclusions. In the future, we intend to augment this explanation capability with even more informative and effective techniques.

12.5.11 User Interface

The user interface (UI) defines the way in which an ES interacts with the user, the environment, and such related systems as databases. The need for an interactive and user-friendly UI cannot be overemphasized and it is deemed to be an important factor in rendering the system easy to learn and easy to use. Indeed, "the interface is critical to the success of any information system, since to the end-user the interface is the system" (Healy et al. 1999). Furthermore, research has shown that interface aesthetics as well as interactivity perform a larger role in users' attitudes than users would admit (Ngo et al. 2001). As such, the perceived usefulness of the interface, or users perception about the usefulness of the interface in a given work domain, plays an implicit role in longer-term user acceptance and performance (Ngo.and Law 2003, Schnecke and Vonberger 1997). Accordingly, we strive for an interactive graphical user interface (GUI) for IdeaL.

Our GUI has the capability to accept input for the layout design from data files saved in text, csv, or Excel format (e.g. dimensions of packing space and modules as well as other parameters). It also has the provision for manual data entry or overriding of preferences from decision makers. Moreover, it enables fast and easy as well as informed and interactive manipulation of layout alternatives by the decision-maker. Some snapshots of Experts' User Interface and Knowledge Acquisition Modules as well as the prototype of end user interface are included in Figs. 12.6 and 12.7 for reference purposes. Details regarding the UI can be found in (Ahmad 2005).

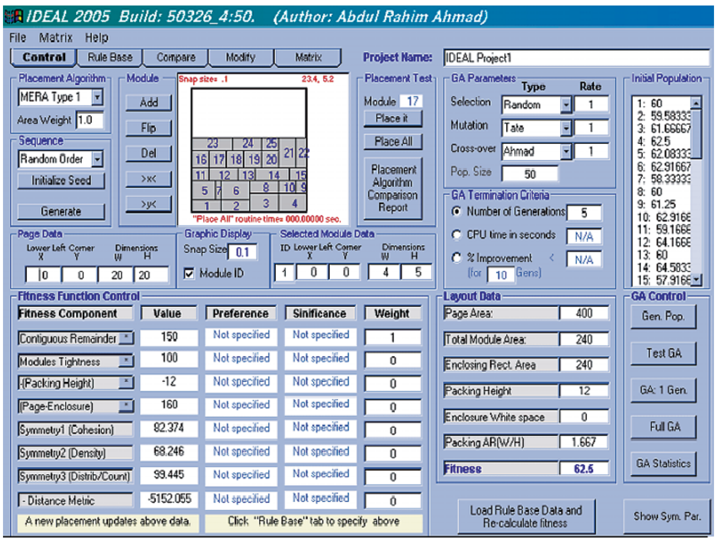


Fig. 12.6. User interface for developers (Normal view)

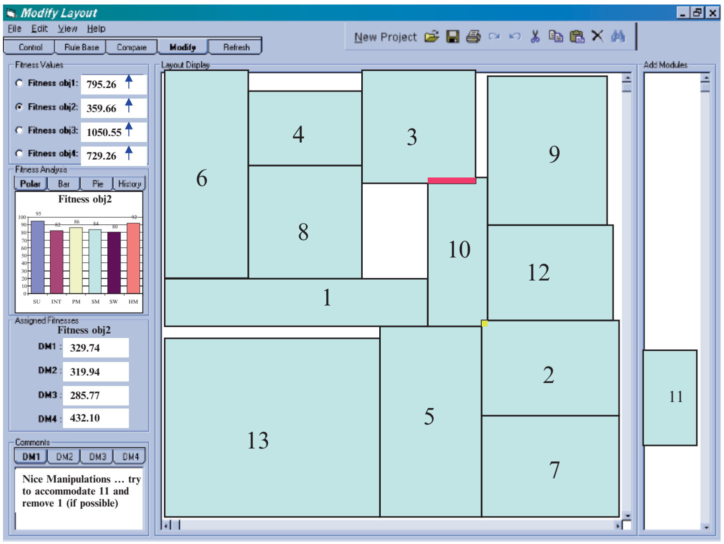


Fig. 12.7. User interface for layout designers

Incidentally, our interface is still evolving. It is because IDEAL is still in the development stage and most of its existing functionalities are designed for developers. Consequently, some of its modules contain a higher degree of complexity to meet ecological requirements of system developers and experts. Indeed, experts operating in complex and dynamic decision-making ecologies prefer to have interfaces that are more complex, nevertheless, powerful (Burns and Hajdukiewicz 2004). However, a prototype of an end-user interface has been developed, and tested, using the philosophy of Ecological Interface Design and various usability and Human-Computer Interaction guidelines (Ahmad 2004). We employed a combination of digital and analog displays for increasing the interface efficacy. Further, our design affords information about the context through various textual, graphical, analogical, and iconic references. Such an interactive interface could become the single most important factor to the eventual success of IDEAL.

Nevertheless, we intend to enhance the usability and interactivity of the interface in the near future. For instance, we could have a window showing one layout and another window showing the modules not included in the layout, enabling the decision maker to move modules in and out of the layout and/or rearrange them in the given layout while simultaneously observing changes in the fitness metrics used to rate that layout. In another mode of interaction, the user might be allowed to see a pair of highly ranked layouts for direct visual comparison and manipulation while observing the changes in fitness values in real time. Some mode of displaying contributions of various determinants of fitness in multi-criteria decision analysis as well as other experts' rating of a layout could augment both interactivity and efficacy of IDEAL.

Indeed, IDEAL's interface affords intervention from decision-makers into the process of generating alternate layouts by modifying membership functions of preferences or weights in the fitness function etc. However, as IDEAL continues to evolve and remove constraints on what could be afforded in its various modes of interaction would furnish creative ways in which they can support decision-makers' work.

12.5.12 Synergy of Intelligent Components

The proposed framework for IDEAL differs from a traditional ES by virtue of various intelligent components. Consequently, we deem it appropriate to elaborate the philosophy and synergic potential of such intelligent components, as these have been the primary focus of this research. This is because of our belief that these components furnish a significant amount of realizable automation in generating and manipulating superior layout alternatives by addressing the core issues in building the whole system. Furthermore, these components furnish a vehicle for carrying out further research in this direction. A somewhat detailed discussion of each intelligent component of IDEAL is provided in the following chapters.

The need for intelligent components arises from limitations of conventional systems design techniques that typically work under the implicit assumption of a good understanding of the process dynamics and related issues. Conventional systems design techniques fall short of providing satisfactory results for ill-defined processes operating in unpredictable and noisy environments such as layout decision analysis and design. Consequently, the use of such non-conventional approaches as Fuzzy Logic (FL), Artificial Neural Networks (ANN), and Genetic Algorithms (GA) is required.

The knowledge of strengths and weaknesses of these approaches could result in hybrid systems that mitigate limitations and produce more powerful and robust systems (Ahmad 2005, Cordon et al. 2004, Negnevitsky 2002). Indeed, the potential of these technologies is limited only by the imagination of their users (Cordon et al. 2004).

Among the intelligent components of IDEAL, *Intelligent Layout Generator* (ILG) generates superior layout alternatives based on pre-specified and user-specified constraints and preferences as well as preference weights furnished by PIA. The *Preference Inferencing Agent* (PIA) incorporates the soft knowledge and reasoning mechanism in the inference engine. The *Preference Discovery Agent* (PDA) complements the ILG and the PIA by automatically discovering and refining some preferences. The proposed synergy is shown in Fig. 12.8.

In this synergy, the PIA receives fuzzy preferences and rules from various sources including domain experts, the knowledge base and the PDA. These fuzzy preferences and rules are defuzzified by the PIA through its inferencing mechanism, furnishing crisp weights for use in the ILG. The ILG, in turn, generates superior layout alternatives for ranking and manipulation by decision-makers. The layout alternatives generated by the ILG could be

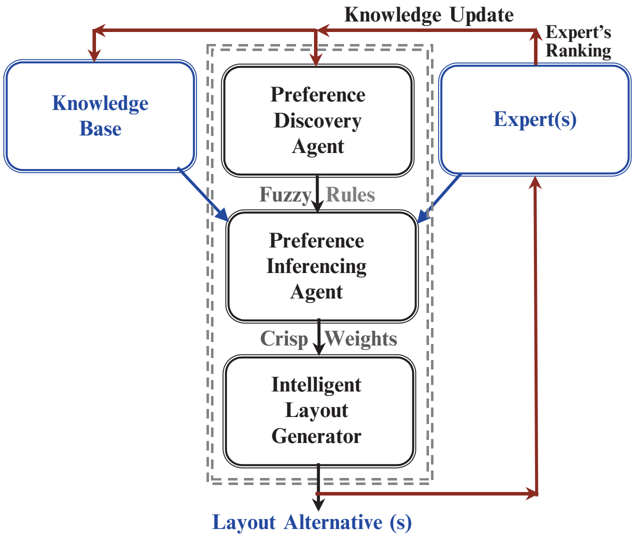


Fig. 12.8. The Synergy of the Intelligent Components in IdeaL

validated by the user or by the PDA. Consequently, the experts' ranking of layout alternatives serve as learning instances for updating and refining the knowledge-base, fuzzy rules, and preferences. Incremental learning technologies like Reinforcement Learning might prove useful here.

These intelligent components combine powers of the three main soft computing technologies representing various complementary aspects of human intelligence needed to tackle the problem at hand (Cordon et al. 2004). The real power is extracted through the synergy of expert system with fuzzy logic, genetic algorithms, and neural computing, which improve adaptability, robustness, fault tolerance, and speed of knowledge-based systems (Ahmad 2005, Cordon et al. 2004, Negnevitsky 2002).

We want to emphasize that these components have deliberately been designed to have a generic character. The rationale behind this philosophy is our belief that a generic approach is more suitable in such subjective, uncertain, and dynamic problem domain as layout design that has applications in a diverse set of work domains. Consequently, a generic approach would result in minimal efforts from design engineer in adapting the system for various layout design problems.

12.6 Bin-Packing Case Studies

Here, we present few test cases to demonstrate the effectiveness of IdeaL and the proposed decision-making paradigm for layout design. Ironically, there is not much literature available on benchmark problems that involve layout design using modules that are unequal in size, fixed in shape, fixed in orientation, and involve subjectivity and uncertainty in placement preferences (Ahmad 2004).

In order to test the viability of IdeaL, we generated several layout alternatives for a 25-module problem using various algorithms. This 25-module problem was procured from a packing industry and has been included in Sect. 12.5.7. These alternatives were given to an expert for getting subjective ratings based on space utilization and layout symmetry as well as any possible manipulation and refinement of those layouts. The expert have more than 20 years of teaching, researching, and practicing experience in layout design applications. The expert neither had knowledge of algorithms used to generate these alternatives nor had any information about the fitness metrics used to evaluate these layouts. Results of those evaluations were used in the training of PDA, as well, as discussed in Sect. 12.5.7. Few interesting instances of this exercise are presented here to demonstrate the efficacy of IdeaL.

Case I. The layout alternative presented in Fig. 12.9 was generated by IdeaL and received a rating a rating of 70 out of 100 from the expert. Apparently, the layout shown in Fig. 12.9 does not seem to be a superior outcome in terms of symmetry or space utilization. However, once again, the higher rating by the expert is a reflection on the fitness potential of the layout alternative following few simple manipulations. It can be seen that the modified topology shown in Fig. 12.10 has higher symmetry as well as space utilization.

It involved the following manipulations: move the module-5 to the bottom-right corner of the bin; move the module-23 on top of modules 5 and 18; move the module-11 to the right of the module-12; move modules 7, 17, and 21 on top of module-23; shift modules 1, 4, and 8 downwards and swap position of modules 1 and 4; move module-14 to the right of module-10. All these nine moves took less than 2 mins. to complete and naturally followed each other.

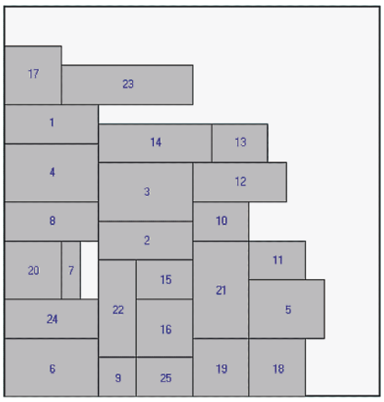


Fig. 12.9. Case I – layout alternative

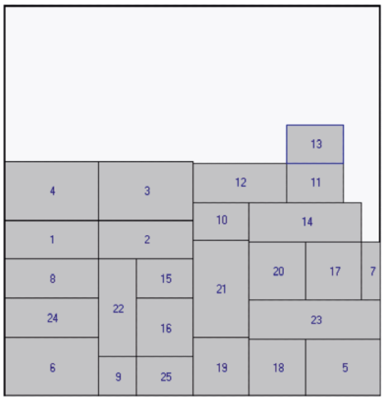


Fig. 12.10. Case I – refined layout

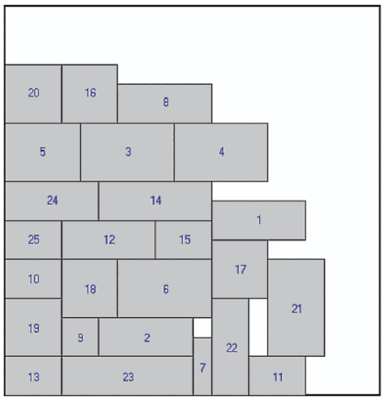


Fig. 12.11. Case II – layout alternative

The resultant layout subsequently received a subjective a rating of 90 out of 100 by the DM.

Case II. The layout alternative presented in Fig. 12.11 was generated by IDEAL and received a rating a rating of 75 out of 100 from the expert. Once again, the higher rating by the expert is a reflection on the fitness potential of the layout alternative following few simple manipulations. It can be seen that the modified topology shown in Fig. 12.12 has higher symmetry as well as space utilization.

It involved the following moves: move module-21 to the right of module-11; move module-17 on top of module-21; move modules 16 and 20 on top of module-21; move module-1 on top of modules 17 and 22; move module-4 on top of module-1; move module-8 on top of module-4. All these six moves took less than one and a half minute to complete and naturally followed each other. When this resultant pattern was given to experts, it received a subjective rating of 85 out of 100.

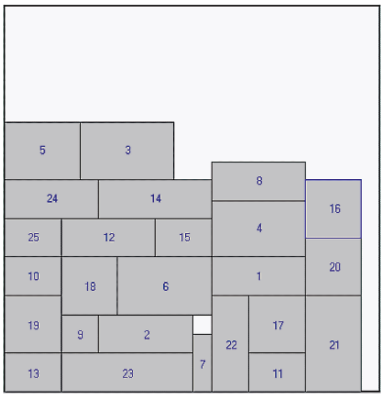


Fig. 12.12. Case II – refined layout

12.7 Future Research

It is hoped that the exclusive and complementary features of various soft computing technologies will result in a synergistic integration, providing new insights to practitioners and theoreticians. Here we list some interesting research directions.

12.7.1 Knowledge Base

Currently, the GA based metaheuristic search approach in IDEAL supports layout design scenarios involving only one bin or packing space. However, the system can be modified to support both multi-bin and undersized bin scenarios. Under such scenarios, some peculiarities may transform the dynamics of the problem and open up some interesting research venues.

In a multi-bin scenario, modules may be placed in a given number of bins, possibly with some effect on the total utility of the layout design. For instance, placement of a particular module on the homepage of an e-Store would have different utility than the case where the same module is placed in one of the subsequent pages.

In an undersized bin scenario, the size of a bin might not be adequate to accommodate all modules. As such, only a subset of modules may be accommodated in a specific layout alternative. In such scenarios, the intrinsic utility of modules as well as inter-module interaction would have more significant role in determining the layout fitness.

12.7.2 Layout Design Heuristics

The need for efficient and effective heuristics in layout design is an ongoing research area where the quest for more useful heuristics would not only facilitate improvements in productivity but also provide more insights to the layout design problem. Heuristics capable of producing solutions with higher aesthetic contents are also important in such subjective problem domains as layout design.

In future, we want to investigate means to facilitate fuzzy placement decisions, such as skipping some less promising placement steps for expediting the design process when the hamming distance between two genes is large. For instance, if the hamming distance between two modules in a chromosome, say A and B, is large then there is little promise in exploring placement of module B at the corners of module A, which are more likely to be occupied already.

12.7.3 Automated Learning

We have demonstrated that automated preference discovery is a pragmatic strategy that offers value in face of difficulty in explicitly articulating preferences by the decision maker. The promise of automated preference discovery provides several potential research streams. For instance, such automatically discovered preferences need to be adjusted or refined based on users' interactions with the preliminary or intermediate alternatives. Explicitly articulating such adjustments in learned preferences by the decision maker might not always be a feasible or an efficient approach. As such, we also need some mechanism to automatically update these preferences. ANN may be used in such an incremental learning mode. However, we believe, few instances of user interactions might not provide sufficient or efficient re-training of the ANN. Consequently, we plan to incorporate a Reinforcement Learning (RL) mechanism for automated updating and refining of preferences and test the viability of automated preference discovery concept under dynamic scenarios.

12.8 Conclusion

In this chapter, we have described the layout design problem, its significance and relevance, and the role intelligent systems and soft computing tools can play in improving the efficacy and efficiency of layout design process. In particular, we have explained the development and working of a novel intelligent approach to solving this important and intricate problem. Our approach involves the use of human intuition, heuristics, metaheuristics, and soft computing tools like artificial neural networks, fuzzy logic, and expert systems. We have explained the philosophy and synergy of the various intelligent components of the system. This research framework and prototype contribute to the field of intelligent decision making in layout design and analysis by enabling explicit representation of experts' knowledge, formal modeling of fuzzy user preferences, as well as swift generation and effective manipulation of superior layout alternatives. Such efforts are expected to afford efficient procurement of superior outcomes and to facilitate the cognitive, ergonomic, and economic efficiency of layout designers as well as future research in related areas.

References

Abdinnour-Helm, S., Hadley, S.W., “Tabu search based heuristics for multi-floor facility layout”, *International Journal of Production Research*, Vol. 38, No. 2, pp. 365–383, 2000.

Adya, S.N., Markov, I.L., Villarrubia, P.G., “Improving Min–cut Placement for VLSI Using Analytical Techniques”, *Proc. IBM ACAS Conf.*, IBM ARL, Feb. 2003, pp. 55–62.

Ahmad, A.R., “A Research Framework for Intelligent Decision Support in the Web Page Layout Design and Presentation Planning”, *M.S. Thesis* (completed but not committed), Management Sciences, Faculty of Engineering, Uni. of Waterloo, April 2002.

Ahmad, A.R., “An Effective User Interface for Knowledge-based Decision Support in Layout Design”, *Working Paper*, Systems Design Engineering, Uni. of Waterloo, 2004b.

Ahmad, A.R., An Intelligent Expert System for Decision Analysis and Support in Multi-Attribute Layout Optimization. Ph.D. Thesis, University of Waterloo, Canada, 2005. http://etd.uwaterloo.ca/etd/arahim2005.pdf

Ahmad, A.R., Basir, O.A., Hassanein, K., “Decision Preferences, Constraints, and Evaluation Objectives in Layout Design: A Review of Modeling Techniques”, *5th Int’l Conf. on Operations and Quantitative Management (ICOQM-V)*, Oct. 2004, Seoul, 2004a.

Ahmad, A.R., Basir, O.A., Imam, M.H., Hassanein, K., “An effective module placement strategy for genetic algorithms based layout design”, *International Journal of Production Research*, Vol. 44, No. 8, pp. 1545–1567, 2006

Akin, O., Dave, B., Pithavadian, S., “Heuristic Generation of Layouts (HeGeL): based on a paradigm for problem structuring”, *Environmental Planning*, Vol. 19, No. 2, 225–245, 1997.

Akoumianakis, D., Savidis, A., Stephanidis, C., “Encapsulating intelligent interactive behaviour in unified user interface artifacts”, *Interacting with Computers*, Vol. 12, pp. 383–408, 2000.

Azadivar, F., Tompkins, G., “Simulation Optimization with Quantitative Variables and Structural Model Changes: A Genetic Algorithm Approach”, *European Journal of Operational Research*, Vol. 113, pp. 169–182, 1999.

Badiru, A., Arif, A., “FLEXPERT: Facility Layout Expert System using Fuzzy Linguistic Relationship Codes”, *IIE Transactions*, Vol. 28, No. 4, pp. 295–309, 1996.

Bazaraa, M.S., “Computerized Layout Design: A Branch and Bound Approach”, *AIIE Transactions*, Vol. 7, No. 4, pp. 432–438, 1975.

Bozer, Y.A., Meller, R.D., “A reexamination of the distance-based facility layout problem”, *IEEE Transctions*, Vol. 29, pp. 549–560, 1997.

Burke, E.K., Kendall, G., Whitwell, G., “A new placement heuristic for the orthogonal stock-cutting problem”, *Operations Research*, Vol. 52, No. 4, pp. 665–671, 2004.

Burns, C.M., Hajdukiewicz, J.R., “*Ecological Interface Design*”, CRC, 2004.

Chazelle, B., “The Bottom-Left Packing Heuristic: An Efficient Implementation”, *IEEE Transactions on Computers*, Vol. 32, pp. 697–707, 1983.

Chung, Y-K., "A neuro-based expert system for facility layout construction", *Journal of Intelligent Manufacturing*, Vol. 10, No. 5, pp. 359–3856, 1999.

Cohoon, J.P., Hegde, S., Martin, W., Richards, D.S., "Distributed genetic algorithms for floor-plan design problem", *IEEE Transactions On CAD*, Vol. 10, No. 4, pp. 483–492, 1991.

Cordon, O., Gomide, F., Herrera, F., Hoffmann, F., Magdalena, Ten years of genetic fuzzy systems: current framework and new trends", *Fuzzy Sets and Systems*, Vol. 141, pp. 5–31, 2004.

Deb, S.K., Bhattacharyya, B., "Fuzzy decision support system for manufacturing facilities planning", *Decision Support Systems*, 2004.

Dowsland, K.A., S. Vaid, W.B. Dowsland, "An algorithm for polygon placement using a bottom-left strategy", *European Journal of Operational Research*, Vol. 141, pp. 371–381, 2002.

Dyckhoff, H., "A topology of cutting and packing problems", *European Journal of Operational Research*, Vol. 44, No. 2, pp. 145–159, 1990.

El-Bouri, A., Popplewell, N., Balakrishnan, S., Alfa, A., "A Search Based Heuristic for Two Dimensional Bin-Packing Problem", *INFOR*, Vol. 32, No. 4, pp. 265–274, 1994.

Fisher, A.M., Nof, S.Y., "FADES: Knowledge-based facility design", *Proceedings of the Int'l Industrial Engineering Conference*, Chicago, May 6–10, pp. 74–82, 1984.

Foulds, L.R., "*Graph Theory Applications*", Springer Berlin Heidelberg New York, 1995.

Garey, M., Johnson, D.,"*Computers and Intractability*", Freeman, NY, 1979.

Gloria, A.D., Faraboschi, P., Olovieri, M., "Block placement with a Boltzman machine", *IEEE Transctions CAD*, Vol. 13, No. 6, pp. 694–701, 1994.

Hassan, M.M.D., Hogg, G.L., "On constructing the block layout by graph theory", *International Journal of Production Research*, vol. 32, No. 11, pp. 2559–2584, 1994.

Head, M., Hassanein, K. "Web Site Usability for eBusiness Success: Dimensions, Guidelines and Evaluation Methods", *Proc. of the Hawaii Int'l Conf. on Business*, 2002.

Healy, P., Creavin, M., Kuusik, A., "An optimal algorithm for rectangle placement", *Operations Research Letters*, Vol. 24, pp. 73–80, 1999.

Heragu, S.S., A. Kusiak, "Machine layout: an optimization and knowledge-based approach", *International Journal of Production Research*, Vol. 28, No. 4, pp. 615–635, 1990.

Hopper, E., Turton, B., "An Empirical Investigation of meta-heuristic and heuristic algorithms for a 2D packing problem", *European Journal of Operational Research*. Vol. 128, pp. 34–57, 2001.

Hower, W., "Placing Computations by Adaptive Procedures", *Artificial Intelligence in Engineering*, Vol. 11, pp. 307–317, 1997.

Irani, S., Huang, H., "Custom Design of Facility Layouts for Multi-Product Facilities Layout Modules", *IEEE Transactions on Robotics and Automation*, Vol. 16, pp. 259–267, 2000.

Islier, A.A., "A genetic algorithm approach for multiple criteria facility layout design", *International Journal of Production Research*, Vol. 36, No. 6, pp. 1549–69, 1998.

Jackson, P., "*Introduction to Expert Systems*", Addison Wesley, England, 1999.

Jakobs, S., "On genetic algorithms for packing of polygons", *European Journal of Operational Research*, Vol. 88, pp. 165–181, 1996.

Karray, F., De Silva, C., "*Soft Computing and Intelligent Systems Design: Theory, Tools and Applications*", Readings, MA: Addison-Wesley, 2004.

Karray, F., E. Zaneldin, T. Hegazy, A.H.M. Shabeeb, E. Elbelgati, "Computational Intelligence Tools for Solving Facilities Layout Planning Problem", *Proc. of the American Control Conference, Chicago*, Vol. 8(4), pp. 367–379, June 2000a.

Kim, J.K., Kwang, H.L.,Yoo, S.W., "Fuzzy bin packing problem", *Fuzzy Sets and Systems*, Vol. 120, pp. 429–434, 2001.

Kumara, S.R.T., Kashyap, R.L., Moodie, C.L., "Application of expert systems and pattern recognition methodologies to facilities layout planning", *International Journal of Production Research*, Vol. 26, No. 5, pp. 905–930, 1988.

Lee, Y.H., Lee, M.H., "A shape-based block layout approach to facility layout problem using hybrid genetic algorithm", *Computers and Ind. Engineering*, Vol. 42, pp. 237–248, 2002.

Leung, T.W., Chan, C.K., Troutt, M.D., "Application of a mixed simulated annealing-genetic algorithm heuristic for the two-dimensional orthogonal packing problem", *European Journal of Operational Research*, Vol. 145, No. 3, pp. 530–542, 2003.

Ligget, R.S., "Automated facilities layout: past present and future", *Automation in Construction*, Vol. 9, 197–215, 2000.

Liu, D., Teng, H., "An Improved BL-Algorithm for Genetic Algorithm of the Orthogonal Packing of Rectangles", *European Journal of Operational. Research*, Vol. 112, pp. 413–420, 1999.

Lodi, A., Martello, S., Vigo, D., "Approximation algorithms for the oriented two-dimensional bin packing problem", *European Journal of Operational. Research*, Vol. 112, pp. 158–166, 1999.

Lodi, A., Martello, S., Monaci, M., "Two-dimensional packing problems: A survey", *European Journal of Operational Research*, Vol. 141, pp. 241–252, 2002.

Love, R.F., Wong, J.Y., "Solving quadratic assignment problems with rectilinear distance and integer programming", *Naval Res. Log. Quarterly*, 23(4), 623–627, 1976.

Mak, K.L., Wong Y.S., Chan, F.T.S., "A genetic algorithm for facility layout problems", *Computer Integrated Manufacturing Systems*, Vol. 11, No. 1–2, pp. 113–127, 1998.

Malakooti, B., Tsurushima, A., "An expert system using priorities for solving multiple-criteria facility layout problems", *International Journal of Production Research*, Vol. 27, No. 5, pp. 793–808, 1989.

Martens, J., "Two genetic algorithms to solve a layout problem in fashion industry", *European Journal of Operational Research*, Vol. 154, pp. 304–322, 2004.

Mazumder, P., Rudnick, E.M., "*Genetic Algorithms for VLSI Design, Layout and Test Automation*", New York, NY: Prentice-Hall 1999.

Meller, R.D., Gau, K.-Y., "The facility layout problem: recent and emerging trends and perspectives", *Journal of Manufacturing Systems*, Vol. 15, 351–366, 1996.

Mir, M., Imam, M.H., "Topology optimization of arbitrary-size blocks using bivariate formulation", *Journal of Computer Aided Design*, Vol. 24, No. 10, pp. 556–564, 1992.

Mir, M., Imam, M.H., "Analytic annealing for macrocell placement problem", *International Journal of Computer and Electrical Engineering*, Vol. 22, No. 2, 1996.

Mir, M., M.H. Imam, "A hybrid optimization approach for layout design of unequal-area facilities", *Computers and Industrial Engineering*, Vol. 39, No. 1/2, pp. 49–63, 2001.

Moon, B.R., Kim, C.K., "Dynamic embedding for genetic VLSI circuit partitioning", *Engineering Applications of Artificial Intelligence*, Vol. 11, pp. 67–76, 1998.

Negnevitsky, M., *Artificial Intelligence: A Guide to Intelligent Systems*", Pearson, 2002.

Ngo, D., "Measuring aesthetic elements of screen designs" *Displays*, 22, pp. 73–78, 2001.

Ngo, D.C.L., Law, B.L., "An expert screen design and evaluation assistant that uses knowledge-based backtracking" *Information & Software Technology*, Vol. 43, 293, 2003.

Norman, B.A., Smith, A.E., "Considering Uncertainty in Unequal Area Block Layout Design", citeseer.ist.psu.edu/271501.html, 2002.

Osman H., Georgy, M., Ibrahim, M., "A hybrid CAD-based construction site layout planning using genetic algorithms", *Automation in Construction*, Vol. 12, pp. 749–764, 2003.

Pierce, J.F., Crowston, W.B., "Tree search algorithms in quadratic assignment problems", *Naval Research Logistics Quarterly*, Vol. 18, pp. 1–36, 1971.

Raoot, A.D., A. Rakshit, "A 'linguistic pattern' approach for multiple criteria facilities layout problems", *International Journal of Production Research*, Vol. 31, pp. 203–222, 1993.

Schnecke, V., Vonberger, O., "Hybrid Genetic Algorithms for Constrained Placement Problems", *IEEE Trans. on Evolutionary Computation*, Vol. 1, No. 4, 266–277, 1997.

Tam, C., Tong, T., Leung, A., Chiu, G., "Site Layout Planning using Nonstructural Fuzzy Decision", *J. of Construction Engg and Management*, May 2002, pp. 220–231, 2002.

Tate, D.M., Smith, A.E., Unequal-area facility layout by genetic search. IIE Transactions 1995, 27: 465–472.

Tommelein, I.D., "SightPlan: a Case Study of BB1", Section 5.5.1 in Sriram, R.D (ed.). *Intelligent Systems for Engineering: A Knowledge-Based Approach*. Verlag, London, 1997.

Tompkins, J.A., White, J.A., Bozer, Y.A., Tanchoco, J.M.A., "*Facilities Planning*", 3rd Ed., Reading: Wiley, NY, 2002.

Turban, E., Aronson, J.E., "*Decision Support Systems and Intelligent Systems*", 6th ed., Reading: Prentice Hall, Upper Saddle River, NJ, 2001.

Walenstein, A., *"Cognitive Support in Software Engineering Tools: A Distributed Cognition Framework"*, Ph.D. Thesis, Computing Science, Simon Fraser University, 2002.

Welgama, P.S., Palitha, S., Gibson, P.R., "Computer-Aided Facility Layout – A Status Report", *International Journal of Advanced Manufacturing Technology*, Vol. 10, No. 1, pp. 66–77, 1995.

Wu, Y., Huang, W., Lau, S., Wong, C., Young, G., "An effective quasi-human based heuristic for solving the rectangle packing problem", *European Journal of Operational Research*, Vol. 141, pp. 341–358, 2002.

Youssef, H., Sait, S.M., Ali, H., "Fuzzy Simulated Evolution Algorithm for VLSI Placement", *International J. on Applied Intelligence, Sp. issue on Applied Metaheuristics*, 2003a.

Zadeh, L.A., "Fuzzy Sets as a Basis for a Theory of Possibility", *Fuzzy Sets and Systems*, Vol. 100 (Supplement), pp. 9–34, 1999.

13

Using Self Organising Feature Maps to Unravel Process Complexity in a Hospital Emergency Department: A Decision Support Perspective

A. Ceglowski[1] and L. Churilov[2,3]

[1] Department of Accounting and Finance, Faculty of Business and Economics, Monash University, Melbourne, Australia
[2] National Stroke Research Institute, Melbourne, Australia
[3] The University of Melbourne, Melbourne, Australia

Summary. In systems that are complex and have ill-defined inputs and outputs, and in situations where insufficient data is gathered to permit exhaustive analysis of activity pathways, it is difficult to get at process descriptions. The complexity conceals patterns of activity, even to experts, and the system is resistant to statistical modelling because of its high dimensionality. Such is the situation in hospital emergency departments, as borne out by the paucity of process models for them despite the continued and vociferous efforts of experts over many years. In such complex and ill-defined situations, it may be possible to access fairly complete records of activities that have taken place. This is the case in many hospital emergency departments, where records are routinely kept of procedures that patients undergo. Extracting process definitions from these records by self organized clustering is neither a pure technical analysis, nor a completely social one, but rather somewhere between these extremes. This chapter describe use of Self Organised Feature Maps to reveal general treatment processes – actual work practices – that may be monitored, measured and managed.

13.1 Introduction

In the 1970s U.S. industries discovered that they had to radically reconsider their approach to business if they were to compete with the high quality imports arriving from emerging economies. Managers had to focus on processes and process control in order to improve quality and productivity. In the intervening decades there has been a great deal of focus on description of business processes yet their elicitation remains as much an art as a discipline. "As is" business processes need to be determined for efficiency and effectiveness assessment, improved decision making and other requirements (Laguna and Marklund 2005). Typically, processes determined to be inefficient or ineffective are either changed (through re-engineering initiatives, for example), or

A. Ceglowski and L. Churilov: *Using Self Organising Feature Maps to Unravel Process Complexity in a Hospital Emergency Department: A Decision Support Perspective*, Studies in Computational Intelligence (SCI) **97**, 365–385 (2008)
www.springerlink.com

modified in order to reduce the variability (through application of initiatives such as Total Quality Management). Decision makers need information about the activities involved in the process, their duration and their sequence both in order to monitor and manage processes for efficiency and effectiveness and in order to direct change. Melan (1993) suggests that process ownership; known boundaries and interfaces; defined process with documented workflow; measurement; monitoring; and control must all co-exist for successful process decisions.

Process ownership is sometimes difficult to determine, being a function of process complexity, culture, organizational structure and so on. The key requisite for process ownership is accountability. This needs to be accompanied by authority over the process (Laguna and Marklund 2005). Ownership carries a high degree of responsibility for decision-making.

Process boundaries are typically points where inputs and outputs cross the system. Internal interfaces exist within processes where jobs are passed from one worker to the next. Most workflow problems are caused by insufficient interface communication (Laguna and Marklund 2005), so interface identification is important to allow focus to be brought to the coordination of activities.

Process definition is achieved by making activities available for review (for instance, as process diagrams). The most common way in which processes are defined is through interviews with experts and people who perform the work (frequently termed the ethnographic approach (Schuler and Namioka 1993)), often as the preliminary step in systems development (Earl 1994; Kotonya and Sommerville 1998; Weerakkody and Currie 2003). Ethnographic approaches are prone to subjective views of the work that may be distorted according to social dynamics unrelated to the work (Rennecker 2004) and may encounter situations where the interviewees are unable to provide generalized pictures of the process (Gospodarevskaya et al. 2005).

Analytical observation may also be used to aid definition of the process. Analysts follow workers and document what they do. This requires the analysts to become embedded in the organization for lengthy periods. As a workflow documentation technique it is better suited to cases where the workflow is largely known and understood.

Finally, existing documentation and data may be analyzed in order to recreate the sequence of activities. This may take the form of referral to operating procedures (that may or may not be followed in practice), or to analysis of records that have been kept of actual work that has been performed. The most rigorous form of this has been termed "Workflow Mining" (van der Aalst et al. 2003) and has its roots in a data mining idea that associations between variables in a relation can be counted and granted some level of confidence (Agrawal et al. 1993). Combination of this concept with inference algorithms (Angluin and Smith 1983) gave a way in which time-series data could be mined (Cook and Wolf 1998) to retrospectively build a picture of sequences of events in software (Agrawal et al. 1998). This idea has been extended so that the branches, loops and joins common in most processes may be inferred from event logs (de Medeiros et al. 2003).

Workflow Mining requires access to a log that records the sequence of defined tasks in workflow for a large number of cases. Such data is readily accessible for most work that takes place within or on computer systems, as evidenced by Business Activity Management software tools for the analysis of logged data such as ARIS-Process Performance Manager (IDS-Scheer 2004). Detailed event logs may be available for computer based systems, but they are seldom available for activities that take place outside computer systems. A wide variety of human-based activities take place outside computer systems. This situation is particularly prevalent in complex environments where (expert) workers make decisions about which activities to implement based on the characteristics of each individual job. This is exactly the situation in emergency departments – data has not been available in sufficient detail to permit Workflow Mining.

To summarize, it may be said that there is a general strategic imperative for businesses to operate efficiently and effectively. One of the most proven ways of doing this is by managing the business processes. Business process management requires the process to be owned, delimited and defined so that it may be measured, monitored and controlled. If the process is neither owned, nor delimited or defined it is unlikely that adequate information will be available for critical management decision making about whether the process is "doing things right" and "doing the right things". If the process is resistant to definition through typical techniques, then effective controls cannot be implemented and the decision maker remains at the mercy of hidden process variations, uncertain outcomes and fluctuating costs.

The objective of this chapter is twofold, namely:

- To present a SOM-based AI methodology for identifying patterns of process activity in the processes that are resistant to definition
- To illustrate the use of this methodology for decision support in a hospital emergency department where work practices are complex and ill-defined

The chapter will first present background to the decision context and describe the complexity of emergency department operations. The difficulty of using conventional process elicitation techniques in this complex environment is described and motivation is provided for the idea of replacing the usual "input-activities-output" perspective of processes with a "patterns of activity" one. This is followed by description of the data requirements, methodology and results of process focused clustering. The clustering results are presented, verified and validated and the implications for decision support in a process environment are discussed prior to concluding the chapter with a general heuristic for the method.

13.2 Decision Context: Process Definition in Hospital Emergency Departments

An Emergency Department (ED) is a hospital department that specializes in providing emergency medical care for patients who are delivered by ambulance, referred by their doctor or choose to seek treatment in an ED. Public hospital EDs provide urgent care to patients with life threatening or serious health problems and also provide care to patients with less serious conditions. Patients with urgent medical needs always take priority.

EDs must be available for patients seeking care, regardless of time of day and number of patients (Duckett et al. 1997). Urban hospital EDs typically draw patients from surrounding residential and industrial areas. The demographic mix of patients is usually wide and can vary owing to intermittent attendance of holiday resorts or sports stadiums. Patients may be of any age and either sex, display a full spectrum of ailments and injuries from life-threatening to minor, and range from lucid to incommunicado (Coleridge et al. 1993; Liaw et al. 2001).

While the progress of patients through the emergency department from arrival to departure can be described at a high level (for instance the series of value-adding functions shown in Fig. 13.1), more detailed models are difficult because of the diversity of symptoms, range of severities and variety of medical specializations involved (Averill et al. 1998; Jelinek 1995b; Walley et al. 2001). Each patient is different in seemingly unpredictable ways so treatment has to be individually customized. Treatment customization may take several forms, from the range of beds that the patients occupy to the tests that are performed, procedures that are implemented and care givers that are involved in treatment (Fig. 13.2). This complexity means that the management of EDs for business (as opposed to clinical) efficiency and effectiveness is challenging – how can one measure performance or variation in a system where no norm exists and every patient is considered unique?

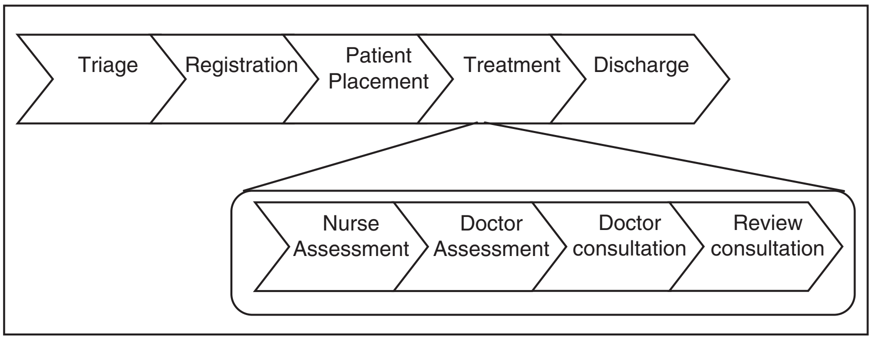


Fig. 13.1. Core value adding functions of EDs (Djohan 2002)

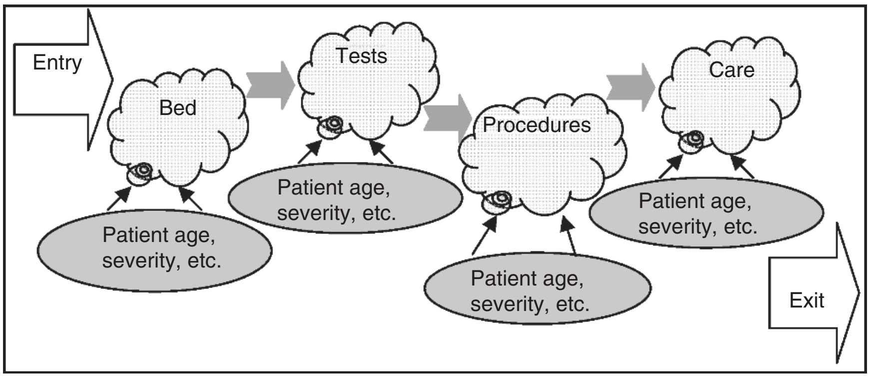


Fig. 13.2. Treatment customization according to patient characteristics

Patient and treatment variability may be dealt with in a number of ways by decision makers wishing to control operations (Harper 2005):

- The variability may simply be ignored. "Average" patients can be used to inform decisions. This approach is particularly risky when dealing with complex systems where net effects are not linear and can lead to poor decisions based on inappropriate assumptions about cumulative factors (Savage et al. 2006).
- At the opposite extreme, each individual patient or treatment may be considered. This view is unlikely to provide any insights that assist decision makers in their tasks. It simply re-creates the complexity of real-life.
- A compromise approach to the two above would be to separate patient characteristics and build distributions of the characteristics. The beds, tests, procedures and care in Fig. 13.2 alone or in combination with patient characteristics may be used to build up theoretical patients by sampling from distributions of each. This approach has some merits in that it encourages the decision maker to evaluate the system from different perspectives (possibly by comparing the average point of each characteristic with its extreme points), and so could contribute to process understanding. However, it could also use or generate "impossible" combinations of characteristics and lead the decision maker astray.
- An alternate approach would be to stream the patients by placing them into fairly homogenous groups. If the streaming were appropriate then the patient groups would have similar process ownership, transition points and process definitions. Decision makers could then measure and monitor the streams and so exercise control over a manageable number of sub-processes.

There have been a large number of initiatives that have attempted the last approach. Simulation (Sinreich and Marmor 2004), industrial engineering and medical casemix concepts (Averill et al. 1998; Cameron et al. 1990; Jelinek

1995b; Walley et al. 2001) are techniques that have been used in an effort to determine how best to stream patients. Models have been attempted that stream patients according to patient characteristics such as age and urgency (Bond et al. 1998; Jelinek 1995a) or on general treatment type (Walley et al. 2001).

Grouping on patient characteristics have only been able to account for some 60 percent of patient related ED costs (Bond et al. 1998). Industrial Engineering approaches that segment patients into different flows such as "simple" and "complex" have improved ED operation to some extent (The Committee on the Future of Emergency Care in the United States Health System 2006), yet EDs remain prone to overcrowding (a situation where there are greater numbers of patients in an ED than it is designed for). Overcrowding can lead to long waits for treatment and complete inability of the ED to accommodate even urgent cases. While patient arrival and departure rates do affect the incidence of overcrowding, patient throughput is also implicated. Patient throughput is affected by the combination of internal ED processes occurring at any given time.

The above analysis shows that even people intimately familiar with ED operations are unable to provide an overview of operations that might be used for management purposes. Without process information decision makers are unable to monitor business operations in a way amenable to proven management practices. This means that EDs are prone to overcrowding and unable to take pre-emptive corrective action when overcrowding threatens.

The decision support problem may now be stated as:

How does one assist management decision making in hospital EDs where little or no process definition exists?

The following sections address the problem by employing a process perspective in data clustering. The data requirements for this are explained in the next section.

13.3 Data Requirements for Process-Focused Clustering

The problem being addressed is how to get a process definition where other methods have not been successful. The ED is the context for this problem. This section will argue that a record of activities that can be used to arrive at adequate process descriptions is quite likely to exist. Groups of activities that describe the core processes may be extracted from the record. The manipulation of the ED data record into a form suitable for process focussed clustering is described prior to discussing the clustering itself in the next section.

Emergency departments keep large amounts of data, but it is collected for medical and legal purposes, not for logging workflow. Doctors and nurses decide on individual phases of patient treatment based on a range of observations. Since patient care takes priority, data may be incomplete and incorrect. Even the advent of electronic patient tracking and electronic (bedside) record maintenance that can generate data logs will only permit EDs with the necessary sophisticated information systems to produce workflow logs. A method is needed that can work within the confines of the current situation.

Fortunately, the "computer external" activities in EDs and other institutions are often logged in databases for billing and other purposes. Such information may be captured batch-wise after the activities have been completed but lack information about sequence or timing of events. Activity logs of this sort are commonly associated with ill-defined processes where experts make complex decisions while performing the work. Recording may be through the selection of activities from multiple-choice entry screens or data entry from paper logs, as is common in hospital emergency departments. If the activities in these records could be analyzed and recurring patterns of activity detected, then the patterns would reflect work practices. The process focused clusters could be used to get an idea of the process variation between sets of activity and the combination of sets of activity that occur.

In the ED context, patients with similar sets of activities would follow similar pathways through the ED and use similar resources. Patient treatments may be determined according to the sets of procedures (the medical analogue of "activities") patients undergo (Fig. 13.3). Complex patients may follow more than one treatment path (consider the case of a person who collapses because of an illness and injures themselves during the collapse – both the injury and the illness need to be treated). This is a simplification of the ED activities involved in patient treatment, but it is nonetheless a record of the actual procedures patients underwent and, consequently, of the treatment sub-processes in the ED.

The data requirements for process definition according to procedures lie somewhere between the knowledge overviews of ethnographic approaches and the time sequenced logs of Workflow Mining. Data is required about procedures, by patient. Data about patient entry and exit times at various points

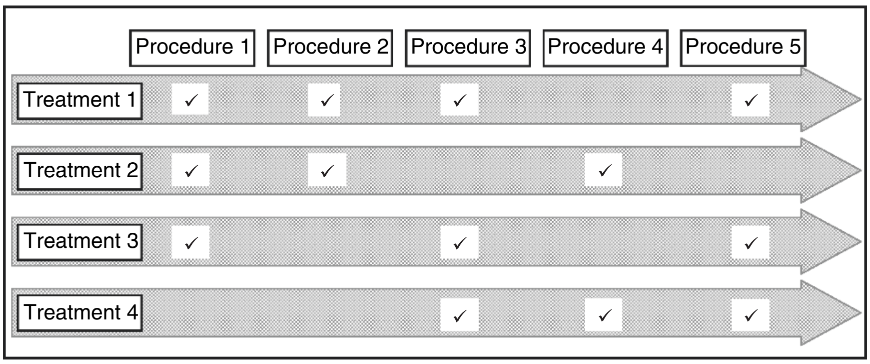


Fig. 13.3. Grouping of procedures into "treatment" clusters. Note that the sequence inferred inthe diagram is illustrative only – precursor/successor information may not be available

would be useful, but not essential, as is patient demographic information such as their age, urgency and so on.

Even though there is a wide range of patients and presentations, much of the work in an ED is based on application of a short list of medical procedures. Patient observation, drug orders, and laboratory and imaging investigations are examples of such procedures. Medical procedures that are listed in data include diagnostic tests such as x-ray or ultrasound; wound and fracture care such as plaster of Paris and dressings; medication and blood product related procedures such as intravenous catheter and fluid; and miscellaneous medical procedures. Others are given in Table 13.1.

Some procedures are much more utilized than others. Of the 68 procedures captured as part of ED reporting, just 36 account for 99% of all procedures in Victorian hospitals. Within this almost 17% are classed as "other" (which includes observation of patients by medical staff); 6% are "No procedures"; some 10% are drug administration and over 9% X-ray imaging. Other significant procedures are venipuncture, intravenous catheter access in preparation for infusion of fluid or drugs, and echocardiogram diagnostics (figures derived from Victorian Emergency Medical Database for 2002).

With the data requirements in mind and the requirements for process focused clustering; patient data made up of 56,906 de-identified records of all emergency department presentations for a year was obtained from a major metropolitan hospital in place. The records contained demographic information plus details of the visit such as key time points and "disposition". The

Table 13.1. Common procedures and their abbreviations

Description	Proc.
Venipuncture	VB
Observation/other	O
Infusion of IV fluid (not blood)	INF
Full ward test of urine	FWT
Computed tomography scan	CT
Head injury observation	HIO
Dressing	DRS
12 Lead ECG + monitoring	ECG
Nebulised medication	NEB
IV drug infusion	IVI
Ultrasound	ULS
Suture, steristrip, glue	SUT
Random blood glucose	RBG
X-ray	XRAY
Drug (oral/sublingual/optical/rectal)	DRUG
ECG monitoring	ECGM
Plaster of paris	POP

This is a subset of the 68 procedures captured for reporting purposes data was cleaned of obvious noise and inconsistencies that related to dates, residence times in the emergency department and errors such as letters in numeric fields.

Data on medical procedures undergone by patients was combined with the records of presentations so that each record contained demographic and visit information plus all medical procedures performed during that visit. The procedures were recorded as integer counts, with zero indicating absence of a procedure. It was possible for a patient to receive repeated applications of a procedure. In practice this was not often the case, except for a generic "observation" procedure which was often repeated. Thus each row of data had an identifier followed by essentially a binary string interjected by the counts between 1 and 5 for the "Observation" procedure variable.

An undirected search for patient treatment groups was initiated so that natural patterns that existed in the data could become apparent. Identification of patient treatment groups is described in the next section.

13.4 Process Focused Clustering

It has been discussed how attempts to summarize ED activities have met with little success because of the complexity of the ED environment. The varieties of patients, range of presentations and scope of treatments have confounded expert efforts to identify commonalities that can adequately encapsulate ED operations in a simple way. The first step in implementing the model of Fig. 13.3 was to determine whether such "treatment" groups existed in the data. There has been almost no research into this issue and expert opinion cannot give definitive direction about what principle pathways patients might follow.[1]

While it may be considered reasonable to use frequent item set algorithms to arrive at common patterns of activity from activity logs (Agrawal et al. 1998; van der Aalst et al. 2003), the focus of frequent item set mining is generally derivation of complete elucidation of all possible combinations of activities that occur in the data. The application of Association Rule algorithms to the frequent item sets provide rules with measures of confidence and support that may be translated to probabilities. There is no indication regarding the interrelationship of clusters with respect to the body of similar instances or to the data set in general.

Clustering algorithms such as Self Organized Mapping (SOM) (Kohonen 1995) provide a likelihood of activities occurring and the clustering solution is usually a "good" rather than optimal or complete one. Various techniques are used to give some assurance that the solution selected is likely to vary as

[1] Clinical pathways, procedure guidelines for the treatment of certain problems, devised by experts through accumulation of best practice knowledge only cover a small proportion of ED treatments, so are inadequate for assessment of ED operations as a whole

little as possible from an optimal one (Orr and Müller 1998). The mapping of clusters into two dimensions by SOM, for instance, provides a visual representation of the size, shape and relative location of clusters in space. Clusters that are adjacent are likely to share inherent properties while clusters that are separated are probably distinctly different. From a process pattern perspective this latter characteristic of two-dimensional maps allows insight into possible meta-grouping of process patterns.

Self-organizing techniques provide an avenue whereby insight can be gained into complex systems through summarizing activities in a comprehensible way; building an understanding of interaction between components; and identifying viable points and measures of control. SOM is a non-parametric clustering technique that makes no assumptions about the distribution of patient characteristics or inter-relationships.

SOM performs clustering well in comparison to a range of other non-parametric methods (Michie et al. 1994). SOM is algorithm-driven and relies on data, rather than domain-specific expertise. It generally employs large data sets, works well with many input variables and is effective in identifying relatively complex cluster models unlimited by human preconceptions (Kennedy et al. 1998).

Availability of tailor-made software and capability of presenting data and analysis results in multiple formats may make SOM more attractive to many analysts than k-mode (Huang 1998), PAM and CLARA (Kaufman and Rousseeuw 1990) which may be more explicit for binary data. The built-in algorithm for optimal number of clusters gives it appeal over k-means where the number of clusters needs to be determined a priori.

SOM and Viscovery SOMine, a software implementation of SOM, have a number of characteristics that suit the identification of process patterns in complex systems:

- The software presents clustering as coloured two-dimensional maps that can quickly be explained to domain experts not versed in clustering methods.
- The inter-relationship of SOM clusters also provide information about meta patterns. The algorithm ensures that adjacent clusters are more related than remote clusters, so further insight may be gained into the interaction of activities that comprise the clusters.
- Activities may occur in more than one cluster so clusters may model parallel processes that utilize common activities, as happens in real life. The type of relationship between activities within clusters cannot be rendered explicitly, but clusters can be recursively mined to learn more about internal relationships.
- It is possible to "overlay" instance characteristics on the map and so learn about the relationships between activities and the outside world.

In the application of SOM to the ED data, records where patients had only one procedure were eliminated from the data set supplied to the clustering algorithm. The ten least common procedures of the 57 in the data were eliminated to bring the number of input variables to less than 50, in line with the software requirement. This involved less than 1% of all records. SOM was then applied to data that comprised of a case identifier and 47 procedures. The clustering results are discussed in the next section.

13.4.1 Clustering Results

The application of SOM to the ED data resulted in 19 clusters that accounted for all patient treatment. The clusters were labelled with the procedures that characterized each cluster to bring the map into the form shown in Fig. 13.4. Note that the labels indicate the primary procedure in the cluster – several other procedures occur within each cluster. Several interesting things may be noted from looking at the map. The first is that the clusters are generally well-formed – they have a regular, roughly circular pattern and the sizes do not differ dramatically. This indicates that the clustering scheme might be "natural" for the data.

The clusters on the right of the map relate to procedures applied in the case of injuries (DRS – dressing; SUT – sutures; POP – plaster of paris; TET – tetanus injection; and so on) and those on the left relate more to investigations related to illness (ABG – arterial blood gases; RBG – random blood glucose; ECG/ECGM – echocardiogram plus monitoring; and so on). Adjacent clusters

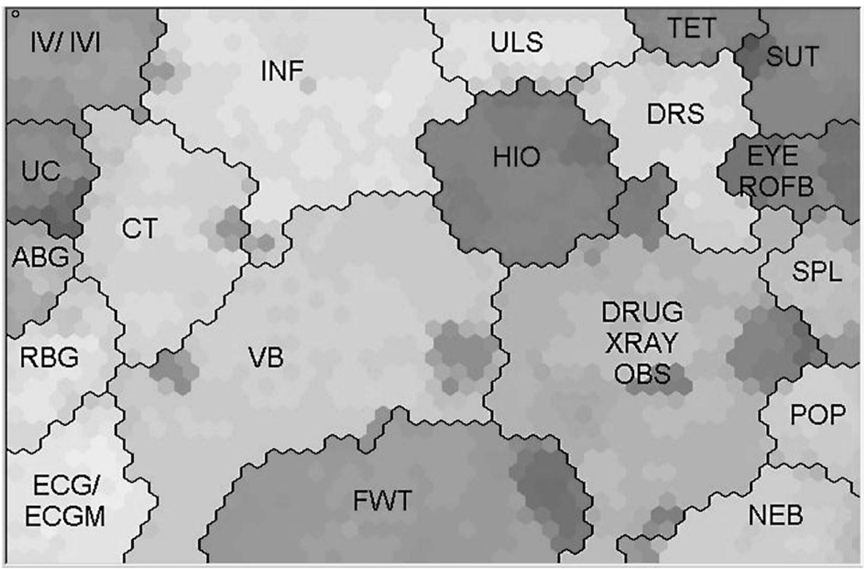


Fig. 13.4. The two-dimensional visual representation of the 19 clusters produced by Viscovery SOMine. The clusters are labelled with the abbreviations for the primary procedures in each

have logical connection. For instance the POP cluster is located close to the XRAY cluster; and SUT is close to DRS and TET clusters (cf. Table 13.1 for abbreviations). Note that these cluster names refer to the primary procedure in the cluster – other procedures are also present, such as X-Rays within the POP cluster.

The NEB cluster (where nebulized medication is used) is placed in the lower right corner of the map, close to injury-related treatments, possibly because respiratory complaints are more similar to injuries in their treatment (rapid application of a limited set of procedures without bedside or laboratory tests) than they are to cases of illness.

After this brief look at the overall map it is pertinent to look more closely at the constituents of the clusters. For the sake of simplicity this will first be presented as a table of the fourteen largest clusters with only the primary procedures indicated (Table 13.2).

Differentiation between clusters may seem trivial if only principal procedures (indicated by "X" in Table 13.2) within each cluster are compared, but the secondary procedures within each cluster provide insight about underlying patterns and similarities between patients in that grouping. It is these

Table 13.2. Fourteen largest treatment clusters for patients who have two or more procedures

	Clusters for patients with two or more procedures													
Proc.	1	2	3	4	5	6	7	8	9	10	11	12	13	14
VB	X		X		0			X		0	0		0	
O	0	X		0	0				0	X				
INF			X											
FWT				X										
CT					X									
HIO						X								
DRS							X					X		
ECG								X						
NEB									X					
IVI										X				
ULS											X			
SUT												X		
RBG													X	
XRAY								0						X
DRUG		0		0										
ECGM														
POP														X
%	14.5	22.0	10.5	8.4	4.2	3.5	3.7	5.8	4.6	4.3	2.6	4.2	2.6	2.0

0: on average, over 60% patients in this cluster underwent this procedure
X: on average, over 80% patients in this cluster underwent this procedure

patterns that supply the necessary information about the overlap of process and clinical activities.

13.4.2 Verification and Validation

Regardless of the method chosen for clustering some effort needs to be expended in ensuring that the results are reproducible and logical. Verification involves checking whether the clustering had been performed correctly and whether the results could be reproduced. Validation determines whether the clusters model the real world.

The cluster quality was assessed to see whether the clustering was "natural" for the data (in other words, whether it could be considered that the solution surface had a significant local minimum in the region of the clustering scheme).

A wide variety of algorithms have been proposed for verifying cluster quality (Han and Kamber 2001; Jain et al. 1999, and many others). The large number of algorithms for determination of cluster quality attest to the fertility of the field, however various algorithms can also give different results owing to different emphasis on either intra-cluster similarity or inter-cluster differences (Bandyopadhyay and Maulik 2001; Bezdek and Pal 1998; Chou et al. 2003). The algorithms generally set out to determine how many clusters are ideal for the data set and whether a defined clustering scheme fits the data set. As such they are the dual problem of clustering itself, which aims to solve the same problems.

Viscovery SOMine suggests optimum numbers of clusters by combining the SOM algorithm with Hierarchical Grouping. Hierarchical Grouping is a form of data grouping where each object or data point is initially a separate "cluster". At each progressive stage the algorithm joins together two clusters that are closest together (agglomeration). The algorithm iterates until a single cluster is formed. Every time a new cluster is formed from the combination of other clusters some level of detail is lost about the individual data points in the cluster (as they acquire the average characteristics of the cluster). Ward (1963) proposed a procedure to quantify this "information loss" as an error of classification calculated at every iteration. Differences in error between iterations give the "step size" of information loss. The optimal number of clusters may be identified at the point of maximum information loss.

The number of clusters can be changed dynamically to experiment with alternative representations. A number of tests were performed to determine whether the clustering was "natural" for the data. These "relative criteria" tests (Theodoridis and Koutroumbas 1999) use a notion of consistent clustering on repeated runs using different parameter settings. Relative criteria tests indicated that 19 was an appropriate number of clusters for the data.

Verification was continued by reviewing Viscovery SOMine's built-in measures of cluster quality. The two-dimensional nature of these quality measures is substantially more powerful than the single index provided by traditional indicators of cluster quality. Every point of the map could be examined for frequency, quantization error, curvature and U-matrix values. While the clustering scheme did not provide the uniformity in measures of the theoretically perfect map, the measures of quality were reasonable in the context of high-dimensional data with numerous records. There were also no unallocated records from the original data set. This means that the clustering scheme was complete for all records in the original data set. This undoubtedly contributed somewhat to the variation of indices seen across the plane of the two-dimensional map.

Validation (whether the clustering scheme seemed an appropriate model of the real world) was initiated by studying the clusters components to see whether they seemed logical. Once it seemed that the clustering was logical from a naïve perspective, the clustering scheme was discussed with the Director of Emergency Medicine at the hospital. By looking at the full output of the clusters he confirmed links between the grouping of procedures and likely presentations by patients. With this support for the clinical relevance of the clustering two last cluster validations were performed.

If the data had carried coded diagnoses then these could have been compared to the clusters to determine whether there was some alignment. Unfortunately this ED did not record diagnoses so this avenue was not available. Other means had to be used. Text mining was carried out on the record of patient symptoms and presentation problem. Good alignment was found between patient symptoms and the procedures indicated in the treatment clusters. Finally, similar clustering was carried out on data from a number of hospital EDs. Similar clusters were found across all campuses.

The verification and validation activities provided reassurance that the grouping of procedures into treatments was accurate enough to consider the treatment processes to be defined. Implications for decision support of the treatment focused clusters are discussed next.

13.5 Implications for Decision Support

This Chapter started with a premise that decision makers who need to manage systems for efficiency and effectiveness cannot exercise adequate control over process without a clear understanding of the interfaces within the process and the activities involved in the process. Knowledge of these components provides *process definition* and facilitates identification of measures by which processes may be monitored and managed.

Unfortunately, in systems that are complex and have ill-defined inputs and outputs, or in situations where insufficient data is gathered to permit exhaustive analysis of activity pathways, it is difficult to get at process descriptions. The complexity conceals patterns of activity, even to experts, and the system is resistant to statistical modelling because of its high dimensionality. Such is

Table 13.3. Summary of the principle arguments of this chapter

	Qualitative approaches	Process focused clustering approach	Quantitative approaches
Processes	Social	Socio-technical	Technical
Data	Interviews and qualitative data	"Activity" records	Complete sequential logs
Process elicitation	Ethnographic	Combination	Algorithmic
Complexity for decision support	High	Moderate	Low
Objective	General understanding of activity flows	Identification of process constituents	Complete enumeration of all possible pathways

the situation in EDs, as borne out by the paucity of process models for them despite the continued and vociferous efforts of experts over many years.

In such complex and ill-defined situations, it may be possible to access fairly complete records of activities that have taken place. This is the case in many hospital EDs, where records are routinely kept of procedures that patients undergo. Extracting process definitions from these records by self organized clustering is neither a pure technical analysis, nor a completely social one, but rather somewhere between these extremes. The clustering algorithm revealed general treatment processes – actual work practices – that may be monitored, measured and managed.

The thinking is summarized in Table 13.3. The processes by which qualitative process elicitation operates are largely social, while that of Workflow Mining are primarily Quantitative. Process focused clustering lies somewhere between these two extremes and strives to identify the most common groups of activities, rather than the more general understanding or exhaustive enumeration of the other methods.

The revealing and confirmation of patterns of activity can have immediate decision support benefits. Decision makers have access to a practical "treatment within urgency categories" overview of what is happening in the ED. Consider the analysis of ED activities by urgency of patients, their treatment, average cycle time between arrival and departure, and the number of patients (Table 13.4). The decision maker can see:

1. Non-exclusivity of treatment by urgency: Patients of different urgency are likely to have similar treatments but the *rate* of application differs. Resource and other implications of increases in the need for certain treatments (as a result of an accident, for instance) can better be estimated by including urgency as a loading factor in the calculations.
2. ED workload may be described as a function of both time that patients receiving particular treatment spend in the ED and the number of patients

Table 13.4. Patient types with the highest cumulative weighted impact on the ED

Urgency	Treatment cluster (typical symptoms)	Disposal	Average ED time in minutes (T)	Number of patients (N)	Weighted impact on ED workload (T × N)
3	3 (intake related vomiting, diarrhoea)	Admit	476	2,003	953,428
3	1 (general malaise)	Admit	438	1,828	800,664
2	8 (cardiac or respiratory)	Admit	524	1,516	794,384
2	1 (general malaise)	Admit	457	1,252	572,164
4	3 (intake related vomiting, diarrhoea)	Admit	505	980	494,900
3	5 (collapse, mental)	Admit	541	877	474,457
3	4 (fever vomiting, diarrhoea)	Admit	445	1,012	450,340
4	1 (general malaise)	Admit	454	909	412,686
4	2 (injury to limb or head)	Discharge	115	3,368	387,320
4	20 (one or no procedures)	Discharge	107	3,538	378,566

The symptoms in brackets implicate hospital wards that might be involved in the admission

receiving that treatment. Table 13.4 indicates that most work at this ED is associated with a narrow range of treatments applied to a large number of patients. Similar analyses by hour of day and season can provide the decision maker with demand profiles and assist in development of measures to deal with specific profiles.

3. The impact of patients awaiting hospital admittance may be clearly shown through their weighted impact on ED workload. The eight highest weighted impact patient types at this ED all related to patients awaiting admission. The large step in average ED time between these patients and those discharged home gives some indication of the efficiencies that might be achieved through faster admission practices. The typical symptoms of patients awaiting admission provide indication of the wards implicated in the delay.

The last point provides a way in which impending blockage may be identified. If excessive numbers of "high workload" patients are in the ED, then the ED is in danger of becoming blocked (Ceglowski et al. 2007). Such a warning system can provide precious time to resource managers to mitigate the impending crisis and perhaps avert ambulance bypass.

This analysis provides some of information available to the process owner about the impact of urgency demand profile and potential blockages to patient throughput (between the ED and certain wards, for instance), but the analyses can be taken to greater levels of detail to provide additional decision support. Individual treatment pathways may be scrutinized to see patterns in patient length of stay, to identify recurring bottlenecks in the system at certain times of the day or days of the week (long waits for sutures on weekends, for instance), or compared to similar analyses from other EDs to arrive at an idea of "best practice".

Treatment groups facilitate use of Fishbone Diagrams (Ishikawa 1986), and other tools of the quality movement because the specific set of medical procedures is known for each treatment. Rather than attempting to build a Fishbone Diagram for every activity in the ED, or being limited to modelling single procedures, analysts can relate groups of procedures in a logical manner. The resulting Fishbone Diagrams are likely to further understanding of each treatment and complexities associated with certain classes of patients (Ceglowski et al. 2004).

It is also possible to allocate costs to the treatment processes, by assembling the component materials costs for procedures. Such costing models reflect the variable costs associated with patient treatment, providing a different perspective to that given by overhead and resource cost allocations and have the potential to lead to full "Casemix" models for ED funding.

Using treatment clusters is possible to analyze which treatments are in process at any given time in the ED, and so describe the combination of procedures that are likely to be required simultaneously. This can be used to build a high level picture of ED operations at any time of day and structure resources accordingly. Tools such as Activity Relationship and Precedence Diagrams that have played a valuable part in location of materials (Francis et al. 1992); layout of workstations (Muther 1973); and balancing of flow lines (Konz 1994), may be applied to aid analysis of these simultaneous and unsynchronised treatment processes.

Treatments can guide the identification of control points, control measures, and the promotion of efficiency and effectiveness – all inaccessible until now because ED work has not been structured along process principles.

The ED example provided in the sections above give some indication of the benefit that might be achieved from a process approach to data clustering. Process focused clustering enables identification of work practices that might not become apparent through other elucidation methods. Identification of process-oriented views of work practices is an essential part of modern decision making and information systems design. The sections above showed how an artificial intelligence technique might be employed to achieve a process-oriented view of ED treatment, the conclusions that follow generalize the approach.

13.6 Conclusions

The suggestion made in this chapter is that self organized methods such as Self Organized Maps supplement other process elicitation approaches. An example was provided of process elucidation in a hospital ED. While self organizing methods may indeed be used on the same data supplied to Workflow Mining, it is specifically proposed to be appropriate when the data is poor in time information. Self organized methods act as an exploratory mechanism to learn more about non-obvious patterns of activity and so help focus process modelling or requirements engineering activities.

It must be realized that data driven methods for deriving process models are unlikely to be sufficient in themselves. They act as a valuable adjunct to ethnographic business process modelling and Workflow Mining methods, adding another tool to the modeller's toolbox.

With the SOM method selected as suitable for the identification of patterns of activity, the following strategy is suggested for the self organization technique of process elicitation:

1. Identify the ill-defined process and build a hypothesis about activities that may provide insight into common patterns of activity.
2. Collect the pertinent data, prepare it and pre-process it.
3. Perform the clustering and validate it on multiple data samples.
4. Interpret the clustering results in terms of process patterns. Activities that are frequently grouped together in clusters probably constitute components of sub-processes.
5. Verify the process patterns through examination of cluster quality.
6. Validate the process patterns to ensure that they reflect the real world (through discussion with experts and observation in the field, for instance).
7. Deploy the process patterns for decision support. This might be done through the use of multiple widely used methods, as described above, or through other means.

Conceptually, this methodology is neither purely qualitative nor quantitative and is aimed at providing practical decision support. In the ED environment and many others, studies and re-engineering projects have been limited to samples of events owing to time and resource constraints. It is not always feasible to have investigators measuring activities every minute of the day over a protracted length of time. The advantage of self organized process focused clustering is that it can use data from every event, or only the data that exists.

In addition to this flexibility in the volume and reach of data, the characteristics that need to exist in the data are less restrictive than those of quantitative workflow methods. Only information about the occurrence of activities is needed whilst start and endpoints and characteristics of the actors are not required.

With these limited requirements it might be expected that the knowledge provided by the method would be highly limited. On the contrary, the artificial intelligence nature of the clustering yields knowledge at several levels, from the size and position of the clusters relative to one another to detailed information about the components within each cluster. The process-orientation of the clustering means that they can be linked to actual work practices. This means that operations can be managed according to assessments of variability, effectiveness and efficiency. The decision maker no longer has to navigate a maze of "individual" events but is able to aggregate events into a reasonable number of logical categories that are amenable to monitoring and analysis.

References

Agrawal, R., Imielinski, T., and Swami, A. (1993) *Mining Association Rules Between Sets of Items in Large Databases*, Paper presented at the ACM SIGMOD Conference on Management of Data, Washington, DC.

Agrawal, R., Gunopulos, D., and Leymann, F. (1998) *Mining Process Models from Workflow Logs*, Paper presented at the Sixth International Conference on Extending Database Technology, Valencia, Spain.

Angluin, D., and Smith, C. H. (1983) *Inductive Inference: Theory and Methods*, Computing Surveys 15 (3) 237–69.

Averill, R. F., Muldoon, J. M., Vertrees, J. C., Goldfield, N. I., Mullin, R. L., Fineran, E. C., Zhang, M. Z., Steinbeck, B., and Grant, T. (1998) *The Evolution of Casemix Measurement Using Diagnosis Related Groups (Drgs)*. 3M Health Information Systems, Wallingford, CT.

Bandyopadhyay, S., and Maulik, U. (2001) *Nonparametric Genetic Clustering: Comparison of Validity Indices*, IEEE Transactions on Systems Man and Cybernetics Part C- Applications and Reviews 31 (1) 120–5.

Bezdek, J. C., and Pal, N. R. (1998) *Some New Indexes of Cluster Validity*, IEEE Transactions on Systems, Man and Cybernetics, Part B 28 (3) 301–15.

Bond, M., Baggoley, C., Erwich-Nijhout, M., and Phillips, D. (1998) *Urgency, Disposition and Age Groups: A Casemix Model for Emergency Departments*, Emergency Medicine Australasia 10 (2) 103–10.

Cameron, J., Baraff, L., and Sekhon, R. (1990) *Case-Mix Classification for Emergency Departments*, Medical Care 28 146–58.

Ceglowski, A., Churilov, L., and Wasserthiel, J. (2004) *Process Mining Informed Industrial Engineering in Hospital Emergency Departments*, Paper presented at the 5th Asia-Pacific Industrial Engineering and Management Systems Conference (APIEMS), Gold Coast, Australia.

Ceglowski, A., Churilov, L., and Wassertheil, J. (2007) *Combining Data Mining and Discrete Event Simulation for a Value-Added View of a Hospital Emergency Department*, Journal of the Operational Research Society 58 (2) 246–54.

Chou, C.-H., Su, M.-S., and Lai, E. (2003) *A New Cluster Validity Measure for Clusters with Different Densities*, Paper presented at the IASTED International Conference on Intelligent Systems and Control, Salzburg, Austria, June 25–27.

Coleridge, J. T., Cameron, P. A., White, J. B., and Epstein, J. (1993) *Profile of Emergency Department Attendances*, Emergency Medicine Australasia 5 (1) 18–25.

Cook, J. E., and Wolf, A. L. (1998) *Discovering Models of Software Processes from Event-Based Data*, ACM Transactions on Software Engineering and Methodology 7 (3) 215–49.

de Medeiros, A. K. A., van der Aalst, W. M. P., and Weijters, A. J. M. M. (2003) Workflow Mining: Current Status and Future Directions. In *On the Move to Meaningful Internet Systems 2003: Coopis, Doa, and Odbase* (Meersman, R., Tari, Z., and Schmidt, D. C., eds.), pp. 389–406. Springer, Berlin Heidelberg New York.

Djohan, V. (2002) *Towards Integrated Clinical Process Modelling in an Acute Emergency Department*, Information Technology, Monash University, Melbourne.

Duckett, S. J., Jackson, D., and Scully, B. (1997) Webpage: *Paying for Hospital Emergency Care*. Acute Health Division, Department of Human Services, Melbourne, http://www.dhs.vic.gov.au/ahs/archive/emerg/index.htm#Contents, Accessed 17 May.

Earl, M. J. (1994) *The New and the Old of Business Process Redesign*, The Journal of Strategic Information Systems 3 (1) 5–22.

Francis, R. L., McGinnis, L. F., and White, J. A. (1992) *Facility layout and location: an analytical approach*. Prentice Hall, Englewood Cliffs, NJ.

Gospodarevskaya, E., Churilov, L., and Wallace, L. (2005) *Modeling the Patient Care Process of an Acute Care Ward in a Public Hospital: A Methodological Perspective*, Paper presented at the HICSS, Hawaii, 3–7 Jan.

Han, J., and Kamber, M. (2001) *Data Mining: Concepts and Techniques*. Morgan Kaufmann, San Francisco.

Harper, P. (2005) *Combining Data Mining Tools with Health Care Models for Improved Understanding of Health Processes and Resource Utilization*, Clinical and Investigative Medicine 28 (6) 338.

Huang, Z. (1998) *Extensions to the K-Means Algorithm for Clustering Large Data Sets with Categorical Values*, Data Mining and Knowledge Discovery 2 283–304.

IDS-Scheer (2004) Webpage: *Aris Process Performance Manager*. IDS-Scheer, http://www.ids-scheer.com, Accessed 13 October 2006.

Ishikawa, K. (1986) *Guide to Quality Control* (2nd revised ed.). Asian Productivity Organization, Tokyo.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999) *Data Clustering: A Review*, ACM Computing Surveys (CSUR) 31 (3) 264–323.

Jelinek, G. A. (1995a) *Casemix Classification of Patients Attending Hospital Emergency Departments in Perth, Western Australia: Development and Evaluation of an Urgency-Based Casemix System*. Doctor of Medicine, University of Western Australia, Perth.

Jelinek, G. A. (1995b) *A Casemix Information System for Australian Hospital Emergency Departments*. A Report to the Commissioner of Health, Western Australia, Perth.

Kaufman, L., and Rousseeuw, P. J. (1990) *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, New York.

Kennedy, R., Lee, Y., Van Roy, B., Reed, C., and Lippmann, R. (1998) *Solving Data Mining Problems through Pattern Recognition*. Prentice Hall, Englewood Cliffs, NJ.

Kohonen, T. (1995) *Self-Organizing Maps*. Springer, Berlin Heidelberg New York.

Konz, S. A. (1994) *Facility Design: Manufacturing Engineering*. Holcomb Hathaway, Scottsdale, AZ.

Kotonya, G., and Sommerville, I. (1998) *Requirements Engineering: Processes and Techniques*. Wiley, New York.

Laguna, M., and Marklund, J. (2005) *Business Process Modeling, Simulation, and Design*. Prentice Hall, Upper Saddle River, NJ.

Liaw, S.-T., Hill, T., Bryce, H., and Adams, G. (2001) *Emergency and Primary Care at a Melbourne Hospital: Reasons for Attendance and Satisfaction*, Australian Health Review 24 (2) 120–34.

Melan, E. H. (1993) *Process Management: Methods for Improving Products and Service*. McGraw-Hill: Copublished with ASQC Quality Press, New York.

Michie, D., Spiegelhalter, D. J., and Taylor, C. C. (1994) *Machine Learning, Neural and Statistical Classification*. Prentice Hall, Englewood Cliffs, NJ.

Muther, R. (1973) *Systematic Layout Planning* (2nd ed.). Cahners, Boston.

Orr, G., and Müller, K.-R. (1998) *Neural Networks: Tricks of the Trade*. Springer, Berlin Heidelberg New York.

Rennecker, J. (2004) Updating Ethnography to Investigate Contemporary Organisational Forms. In *The Handbook of Information Systems Research* (Whitman, M. E., and Woszczynski, A. B., eds.), pp. xi, 349. Idea, Hershey, PA.

Savage, S., Scholtes, S., and Zweidler, D. (2006) *Probability Management*, OR/MS Today 33 (1).

Schuler, D., and Namioka, A. (1993) *Participatory Design: Principles and Practices*. Lawrence Erlbaum, Hillsdale, NJ.

Sinreich, D., and Marmor, Y. N. (2004) *A Simple and Intuitive Simulation Tool for Analyzing Emergency Department Operations*, Paper presented at the 2004 Winter Simulation Conference, Washington, DC.

The Committee on the Future of Emergency Care in the United States Health System (2006) *Hospital-Based Emergency Care: At the Breaking Point*. Institute of Medicine (IOM), Washington, DC.

Theodoridis, S., and Koutroumbas, K. (1999) *Pattern Recognition*. Academic, San Diego.

van der Aalst, W. M. P., van Dongena, B. F., Herbst, J., Marustera, L., Schimm, G., and Weijters, A. J. M. M. (2003) *Workflow Mining: A Survey of Issues and Approaches*, Data & Knowledge Engineering 47 (2) 237–67.

Walley, P., Watt, A., Davies, C., Huang, A., and Ma, K. (2001) *A Study of Demand for Emergency Access Health Services in Two UK Health Regions*. Warwick Business School, Warwick.

Ward, J. (1963) *Hierarchical Grouping to Optimise an Objective Function*, Journal of the American Statistical Association 58 236–44.

Weerakkody, V., and Currie, W. (2003) Integrating Business Process Reengineering with Information Systems Development: Issues and Implications. In *Business Process Management (Bpm2003)* (van der Aalst, W. M. P., ter Hofstede, A. H. M., and Weske, M., eds.), pp. 302–20. Springer, Berlin Heidelberg New York.

14

Future Directions: Building a Decision Making Framework Using Agent Teams

Jeffrey Tweedale[1], Christos Sioutis[2], Gloria Phillips-Wren[3], Nikhil Ichalkaranje[4], Pierre Urlings[2], and Lakhmi C. Jain[1]

[1] School of Electrical and Information Engineering, Knowledge Based Intelligent Engineering Systems Centre, University of South Australia, Mawson Lakes, SA 5095, Australia
[2] Air Operations Division, Defence Science and Technology Organisation, Edinburgh SA 5111, Australia
[3] The Sellinger School of Business and Management, Loyola College in Maryland, Baltimore, MD 21210 USA
[4] School of Electrical and Information Engineering, University of South Australia, Adelaide, South Australia SA 5095, Australia

Summary. This chapter describes initial efforts and research directions in decision support systems that allow collaboration and cooperation between intelligent agents in a multi-agent system and humans. Description of previous research is included to show how developments in the agent software framework was implemented based on cognitive hybrid reasoning and learning models where decision support systems are used to support the human's roles. Cooperation is a type of relationship within structured teams when an agent is required to coordinate with, and explicitly trust, instructions and information received from controlling agents. Collaboration involves the creation of temporary relationships between different agents and/or humans that allow each member to achieve his own goals. Due to the inherent physical separation between humans and agents, the concept of collaboration has been identified as the means of realizing human-agent teams to assist with decision making. An example application and preliminary demonstration to show the current status is also presented. Future research needed to advance the field of intelligent decision support systems is identified.

14.1 Introduction

Decision Support Systems (DSSs) emerged in the early 1970s to assist and support humans in the decision making process. DSSs were initially generated by computer programmers in an attempt to capture the knowledge of subject matter experts in an information management system that could ideally be used to assist management in making decisions without the need for consultation or detailed analysis. The number of applications has expanded as

J. Tweedale et al.: *Future Directions: Building a Decision Making Framework Using Agent Teams*, Studies in Computational Intelligence (SCI) **97**, 387–408 (2008)
www.springerlink.com

computers have become ubiquitous and essential in professional and personal tasks. Recent advances in Artificial Intelligence (AI) have provided a new set of techniques and methods for DSSs that increase their scope and effectiveness. The chapters in this book attest to the intriguing possibilities of Intelligent Decision Support Systems (IDSSs) as combinations of DSSs and AI techniques to effectively support human decision making in complex environments. In this chapter we discuss some of the potential future developments in IDSSs.

One of the more promising areas of AI research for incorporation in IDSSs is intelligent software agents (or just agents). As indicated by Russell and Norvig (2003), an agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors. Agent-oriented development can be considered as the successor of object-oriented development when applied in AI problem domains. Agents embody a software development paradigm that attempts to merge some of the theories developed in AI research within computer science. Bratman's Beliefs, Desires, Intentions (BDI) reasoning model (Bratman 1999) has demonstrated the potential of becoming the method of choice for realizing truly autonomous agents. *Beliefs* represent the agent's understanding of the external world; *desires* represent the goals that it needs to achieve; and *intentions* are the courses of action that the agent has committed to follow in order to satisfy its desires (Rao and George, 1995).

When defining the intelligence of agents, researchers generally state the properties that a system of agents should exhibit. Firstly, *autonomy* means operating without the direct intervention of humans. Secondly, *social ability* means interacting with other agents. Thirdly, *reactivity* means perceiving their environment and responding to any changes that occur in it. Finally, *pro-activeness* means exhibiting goal-directed behavior (Wooldridge 2002). The social ability of agents provides the potential to create stand-alone or cooperative agents that communicate with other agents as required. Different techniques have been developed allowing agents to form teams, and agents can be dynamically assigned a particular role depending on the situation and their suitability. Recent advances in this field have focused on the formation of rather unique teams with human and machine members based on cognitive principles. One major advantage of such teams is an improved situation awareness capability for the human when dealing with unknown or hostile environments (Urlings, 2003).

This chapter focuses on the design of intelligent agent architectures. Agent teaming ability is illustrated with a simulation environment relevant for Airborne Mission Systems. Agent teaming has gained popularity in recent years and is categorised into the prominent domain of Multi-Agent System (MAS). It is believed that three important aspects, 'Communication, Coordination and Cooperation', play an important role in agent teaming. Multi-agent teaming takes inspiration from human organisational models of team operation, where leadership, communication, cooperation and collaboration skills empower the success of the team. In addition, future research directions and needs are identified.

14.2 Models of Decision Making

Information overload occurs when the amount of information available to the user for decision making is more than can be processed in a relevant time period. It is often associated with real-time decision making in which information changes rapidly, the quantity of information is large, and the relationships between the data items are difficult to discern. The Observe – Orient - Decide - Act (OODA) loop, also know as the four box method, shown in Fig. 14.1, is one approach used to aid humans in making decisions when overloaded with information. The cycle was originally labeled by Boyd as the OODA loop to assist pilots, as military decision-makers, to achieve knowledge superiority and avoid information overload in order to win the battle (Coram 2002). Boyd studied air-to-air engagements of the Korean War (1950–1953) in which US fighter pilots, despite flying F-86 Sabre aircraft with wider turn radii, had a consistent 10:1 victory ratio over MiG-15 aircraft that had much better manoeuvrability.

While conventional wisdom suggested that US pilots were successful because they were better trained, Boyd suspected it was due to much more. His hypothesis was that a US pilot would win almost every dogfight because he could complete loops of decision-making much faster than his adversary. Boyd constructed such a loop with the four distinct steps shown in Fig. 14.1 (Curts and Campbell, 2001):

Observe - US pilots could see their adversaries earlier and better because the cockpit design of their aircraft ensured better visibility.

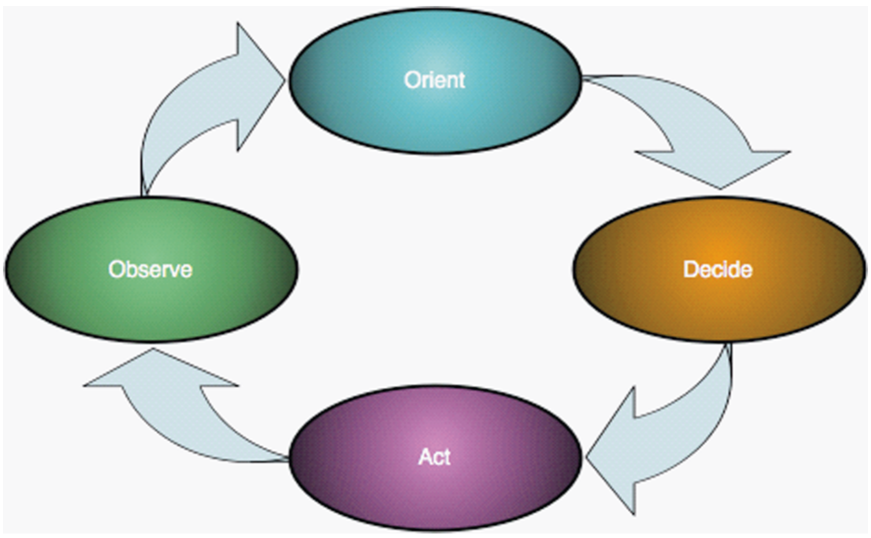


Fig. 14.1. Boyd's observe-orient-decide-act loop

Orient - Since the adversary was acquired first, US pilots could then react by orienting themselves towards the adversary much faster.

Decide - After reacting with their initial orientation, the better level of training then allowed them, as decision makers, to proceed faster to the next combat manoeuvre.

Act - With the next combat manoeuvre decided upon, US pilots could then rapidly input aircraft controls, with the resultant faster initiation of a desired manoeuvre (the F-86 Sabre was more nimble than the MiG-15 because of its fully hydraulic controls).

Boyd conceptualised the principles of the OODA loop in his two famous briefings "patterns of conflict" and "a discourse on winning and losing", which are considered the most dazzling briefings ever to come from a military mind. These presentations began as one-hour and grew to fifteen-hour briefings over two days and were given over 1,500 times. Thousands of copies have penetrated US military and defense circles, particularly at senior levels. Boyd never formally published his observations, but he has been recognized as the greatest military theoretician since Sun Tzu and as the architect of America's strategy in the 1990–1991 Gulf War (Coram 2002, Hammond 2004). The OODA loop has become a standard model of the decision-making cycle not only for the military, but also by many business and research communities around the world (Hammond 2004).

In comparison, Noble Prize winner Herbert Simon studied management decision making and developed a more generalized model of decision making (Simon 1977). Simon's model is shown in Fig. 14.2 with four phases (the final phase added by later researchers) of Intelligence – Design – Choice – Implementation. During the intelligence phase, the user seeks and acquires information needed for the decision problem. Design involves developing criteria important to the decision and establishing relationships between variables of interest. The user makes a selection during choice, and the decision is implemented during the final phase. The phases proceed relatively sequentially, with feedback loops as the user returns to a previous stage before moving forward again. Boyd's model and Simon's model both involve feedback loops and are similar in that the first phase involves acquiring information, the second developing a model to relate the information, the third making a choice, and the fourth acting on the information.

14.3 Intelligent Decision Support Systems

Incorporating AI techniques within DSSs to form IDSSs is not new. However, recent advances have enabled better accessibility to AI technology that has resulted in an increased number of IDSS applications, particularly those using multi-agent systems. These types of applications can aid the decision maker in selecting an appropriate action in real-time under stressful conditions by enabling up-to-date information, reduced information overload,

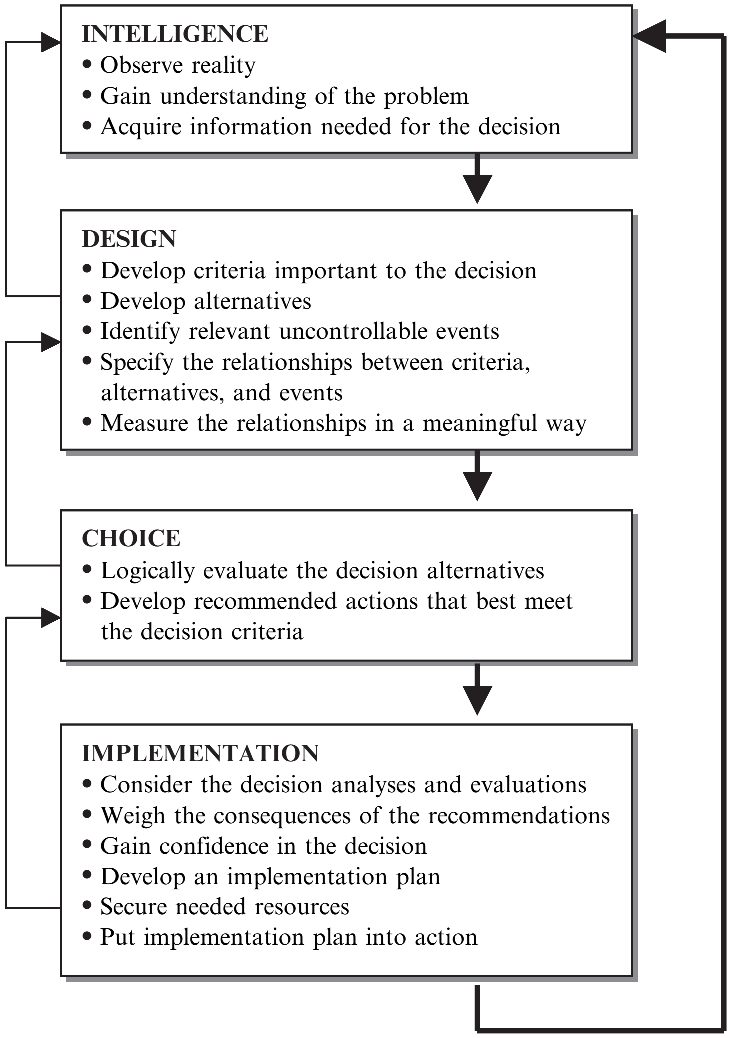


Fig. 14.2. Simon's three phases of decision making, with the last phase added later

and a dynamic response. Intelligent agents can be used to enable communication required for collaborative decisions and to treat uncertainty in the decision problem. AI researchers possess a comprehensive toolbox to deal with issues such as architecture and integration (Mackworth 2005). Several recent examples include:

Petroleum production: Based on Case Based Reasoning (CBR) using bioinformatics (Bichindaritz and Marling 2006, Chan 2005);

Clinical healthcare: Using collaborative decision making and knowledge exchange (Frize et al. 2005);

Forest fire prevention: Based on fuzzy modeling (Iliadis 2005);
Diagnosing breast cancer: Using Linear Genetic programming (LGP);
Multi Expression Programming (MEP) and Gene Expression programming (Jain 2000).

Intelligent Agents (IA) are perhaps the mostly widely applied AI method in IDSSs in recent years due in part to their characteristics of mobility and autonomy. This utilization has significantly advanced many applications, particularly Web-based systems (see for example, Phillips-Wren and Jain 2005). In addition, learning can be incorporated into agent abilities to extend the capability of systems (Valluri and Croson 2005).

14.3.1 Agent Teaming

An agent-enabled IDSS can be designed using a multi-agent system to provide simultaneous data analysis and to enhance the fidelity of feedback to the user. The architecture of the proposed system resembles that of a simple thermostat, containing a monitor and a feedback circuit. Building blocks of this type lead to expert systems and the creation of production rules in the form of logical expressions to embody knowledge. These rules are entered into the knowledge repository as a set of inferences. MYCIN (Simon 1977) and DENDRAL (Feigenbaum et al. 1971) were early commercial versions of DSSs using an expert system as its source of knowledge/inference. An IDSS that uses a multi-agent system to monitor and log the environment prior to deciding on the type and amount of feedback requires significant planning. To interact, the system needs to react to changes at its input from a sensor (using an event-driven model) and produce outputs to drive actuators (again using an event-driven model). These agents can be instantiated using off-the-shelf expert system shells (Negnevitsky 2005). This means that knowledge needs to be represented in terms of rules generated by a subject mater expert prior to use. Such rules should be expressed in terms of Relationships, Recommendations, Directives and Strategies. Separate agents are generally used to collect and refine the test data required to build and test the system. An additional interface agent (or team of agents) is used to interface the inference engine and another agent (or team of agents) to generate feedback and reports.

There are three primary challenges that must be overcome to effectively form agent teams: Communication, Negotiation, and Trust. Communication is concerned with the means of communication between agents such that they can understand each other. Early agent development relied on the idea that intelligence is an emergent property of complex interactions between many simple agents. For example, the Open Agent Architecture (Cheyer and Martin 2001) is based on agents in a community of agents cooperating to achieve their design objectives. Communication between agents must be efficient and robust enough to recover easily from errors, and specialized Facilitator agents are responsible for matching requests with the capabilities of different agents.

Another approach is given by Aglets (Lange 1997) as Java objects that can move from one host on the network to another. Such mobile agents are particularly useful in distributed systems. Finally, Swarm (Group 2005) provides a hierarchical structure that defines a top level observer swarm; a number of model swarm are then created and managed in the level below it.

The second challenge to forming agent teams is Negotiation. Generally, development of teams involves separating the requirements of a team from the requirements of individual agents. This includes assigning goals to the team as a whole, and then allowing the team to figure out how to achieve it autonomously. A team is constructed by defining the number of roles that are required in order to achieve the goals of the team. Additionally, agents can be specifically developed to perform one or more roles. An important feature of this approach is that agents are assigned with roles at runtime and can also change roles dynamically as required. Hence, one agent may need to perform one or more roles during its operation. MadKit (Ferber et al. 2006) is a multi-agent platform built upon an organizational model called Agent/Group/Role, and agents may be developed in many third party languages. The widely used 'JACK Teams' (AOS 2006) provides a team-oriented modeling framework. Specifically, this allows the designer to focus on features such as team functionality, roles, activities, shared knowledge and possible scenarios.

The third major challenge to agent team formation is Trust, specifically how an agent should handle trust in regards to other agents. For example, should an agent trust the information provided by another agent, or trust another agent to perform a particular task. The level of trust is not easily measured, although loyalty can be used to weight information and consequently the strength of bond that is created. The fragility of that bond reflects on the frequency and level of monitoring required for the team to complete the related portion of a task. For further details on trust, the reader may refer to Tweedale and Cutler (2006).

One would expect to gain major benefits from intelligent agent technology through its deployment in complex, distributed applications such as virtual enterprise management and the management of sensor networks. However, while the agent paradigm offers the promise of providing a better framework for conceptualising and implementing these types of systems, there is a need to recognise the underlying programming paradigms and supporting standards, design methodologies and reference architectures needed before these applications can be developed effectively. As noted above, standards are beginning to appear, but more experience and is needed with real applications, and the software community needs to be educated in their use. Given the nature of these applications, a sudden shift to an agreed-upon standard in the community seems unlikely. Rather, as the field matures we would expect to see a gradual shift from object-oriented to the agent paradigm in intelligent domains.

The underlying theories of cognition will continue to prove adequate for large-scale software developments. The key theories (BDI and production systems) date from the 1980s and have a long pedigree in terms of their use in commercial-strength applications. This longevity indicates that their basic foundation is both sound and extensible as clearly illustrated in the progression of BDI implementations from PRS (Francois et al. 1996) to dMARS (d'Inverno et al. 1997) to JACK (AOS 2005) and to JACK Teams (AOS 2006). New cognitive concepts may gain favour (e.g. norms, obligations, or perhaps commitment), but we believe that these concepts will not require the development of fundamentally new theories.

While we believe that the existing theories are sufficiently flexible to accommodate new cognitive concepts, we perceive a need to develop alternative reasoning models. In the case of the JACK implementation of BDI, a team reasoning model is already commercially available in addition to the original agent reasoning model. On the other end of the spectrum, a low-level cognitive reasoning model (COJACK) has been recently developed. This model enables the memory accesses that are made by a JACK agent to be influenced in a cognitively realistic manner by external behaviour moderators such as caffeine or fatigue. Interestingly, COJACK utilises an ACT-R like theory of cognition, which in turn is implemented using JACK's agent reasoning model. From a software engineering viewpoint, it should be the reasoning model that one employs that shapes an application, not the underlying cognitive theory. There is the opportunity through the provision of "higher level" reasoning models like OODA and their incorporation into design methodologies to significantly impact productivity and, hence, market penetration of these technologies.

14.3.2 Collaborating Agents to Simulate Teamwork

A number of researchers have integrated cognitive decision-making models with agents (Klein 1989b, Yen et al. 2001) to capture the decision making abilities (Klein 1989a) of domain experts based on the recognition of similarity between the current situation and past experiences. In the first (recognition) phase, a decision maker develops situation awareness and decides upon a course of action. In the second (evaluation) phase, a decision maker evaluates each course of action. Klein (1989a, b) introduced a model that evolved into an agent environment under teamwork setting into the Recognition-Primed Decision (RPD) Agent architecture (Fan et al. 2005b). Klein's cognitive model was extensively tested in highly stressful, time-pressured, decision making environments such as those faced by firefighters or military personnel under attack. He proposed that these types of decision makers base their responses on past experience and situations that are similar to the new situation. Hanratty et al. showed an agent architecture for a RPD Agent (Hanratty et al. 2003) as consisting of four modules. The communication manager module governs the inter-agent communication and organises conversations. The expert system module is a rule-based forward chaining system containing knowledge related to the other agents and external world. The process manager module is responsible for scheduling and execution of plans. The collaborative module facilitates the collaboration between humans and RPD agents (Klein 1989a).

Software called Recognition-Primed Collaborative Agent for Simulating Teamwork (R-CAST) was developed based on the RPD model using similarities between past experience and current situation. The Pennsylvania State University has filed a patent on the software embodied in R-CAST, an extension of the Collaborative Agent for Simulating Teamwork (CAST) architecture (Fan et al. 2005a). CAST was designed to simulate teamwork by supporting proactive information exchange in a dynamic environment, while R-CAST extended CAST architecture with a recognition-primed decision making model. R-CAST consists of a number of modules for handling the collaboration among RPD-agents, between RPD-agent and human, and among humans. The Shared Mental Model (SMM) consists of team processes, team structure, shared domain knowledge, and information-needs graphs. The Individual Mental Model (IMM) stores mental attitudes held by agents. Information is constantly updated using sensor inputs and messages from agents. The Attention Management (AM) module is responsible for the decision-maker agent's attentions on decision tasks. The Process Management (PM) module ensures that all team members follow their intended plans. The functions of the other modules are described by Yen et al. (2001).

The developers of R-CAST and RPD Agent have tested their software in a military command-and control simulation involving intelligence gathering, logistics and force protection (Hanratty et al. 2003). Under normal time pressure, the human teams made correct decisions about the potential threat. As time pressure increases, team performance suffers due to the lack of information sharing resulting in incorrect decisions about whether to attack/avoid the incoming aircraft. The researchers demonstrated that the R-CAST agent systems helped human-agents in making the right decisions under time-pressured conditions. This concept is demonstrated using a scenario in which team members have to protect an airbase and supply route that are under attack by enemy aircraft. The scenarios were configured with different patterns of attack and at different tempos. Two human team members were dependent on a third human whose role was to gather information and communicate to them. The defence teams cannot attack if they do not know whether the incoming aircraft is friend or foe. The supply team takes action to avoid a possible incoming threat. When the information gatherer was supported by the R-CAST software system, the information was processed and shared quickly. As a result, the human-agent teams were able to defend themselves from enemy attack.

14.3.3 JACK Intelligent Agents

JACK Intelligent Agents is a development platform for creating practical reasoning agents in the Java language using BDI reasoning constructs. It allows the designer to use all features of Java as well as a number of specific agent extensions. Any source code written using JACK extensions is automatically compiled into regular Java code before being executed. Each agent has beliefs about the world, events to respond reactively, goals that it desires to achieve, and plans that define what to do. When an agent is executed, it waits until it is provided with a goal to achieve or receives an event to which it can respond reactively; it then reasons using its beliefs and decides whether to respond. If a response is required, it selects an appropriate plan to execute in order to respond. JACK agents can exhibit: Goal-directed behavior, where the agent focuses on the objective and not the method chosen to achieve it; Context sensitivity, keeping track of which options are applicable at each given moment using beliefs; Validation of approach, ensuring that a chosen course of action is pursued only for as long as applicable; and Concurrency, behaviours in the agent are executed in separate, parallel and prioritized threads.

JACK provides a language for developing agent-based systems using agent-oriented paradigms, and the language is complete with a compiler, a powerful multi-threaded runtime environment and a graphical environment to assist with development. Beliefs have been implemented as relational databases called beliefsets; however, developers can also use their own Java-based data structures if needed. Desires are realized through goal events that are posted in order to initiate reasoning. This is an important feature because it causes the agent to exhibit goal-directed behaviour rather than action-directed behaviour, meaning that the agent commits to the desired outcome and not on the method to achieve it. An intention is defined as a plan to which the agent commits to after choosing from a library of pre-written plans. The agent is able to abort a plan at any time depending on its beliefs and also consider alternative plans.

JACK Teams is an extension to the JACK platform that provides a team-oriented modelling framework. The JACK Teams extension introduces the concept of Team reasoning, where agents encapsulate teaming 'behaviour and roles' required to define what each agent is required to do within the team. Using this Teams extension of JACK, individual agent functionality is also available within a team. Team-oriented programming enables the designer to specify: What functionality a team can perform; What roles are needed in order to form a team; Whether an agent can perform a particular role within a team; Coordination of activities between team members; Knowledge between team members.

Roles are bound to agents at runtime. This means that it is possible to have different combinations of agent-role relationships. For example, on one hand, one role can be performed by many different agents (in which case one agent must be selected at runtime), on the other hand, one agent can also perform many roles simultaneously as required.

Belief propagation allows beliefs to be shared between members of a team. This means it becomes possible for sub-teams to inherit beliefs with important information from higher-level teams and conversely, enclosing teams to synthesize beliefs from lower-level sub-teams. JACK Teams was developed to support structured teams; therefore, the role obligation structure of a team must be defined at compile-time. Consequently, sub-teams can only communicate and share information if it has been previously defined in their team structure.

14.3.4 Teaming

The team in this concept can initially be considered to consist solely of software agents. However, ultimately the team will include human agents or operators in either a collaborative or commanding mode. The communication aspects in agent teaming address traditional teaming properties such as exchange of information as well as agent and mutual performance monitoring. Research focus is needed in communication and collaboration between software and human agents.

The structure of teams is traditionally defined during the system design and is required to remain constant during operation. Within teams, agents are required to cooperate and explicitly trust other team members. The idea of introducing dynamic, temporary team-like links that can be established or destroyed at runtime also needs to be considered. This approach allows the achievement of greater autonomy since different systems, each executing different agent teams, are able to collaborate in order to achieve their goals. Additionally, agent teaming should be considered to contain a 'human-centric' nature. Current research trends in agent development needs to focus on how agents interact within teams.

One of the major issues in early human-machine automation was a lack of focus on human users and their cognitive processes. Recent developments in intelligent agents have become a popular way to respond to these early deficiencies. Early agent models or theories were attractive solutions due to their human-like intelligence and decision-making behaviour. Existing agent models can act as stand-alone substitutes for humans and their human decision-making behaviours.

At this point we come back to one of the problems in early human-machine automation – the human-like substitute could fail at a critical point due to cultural diversity or lack of coordination, leaving the human no chance to regain control of the situation (usually as a result of impaired situation awareness). A solution was developed by AI researchers who created a machine-assistant operating in an advisory or decision support role and that assisted human operators during critical or high workload situations. This software led to the development of intelligent agent technology. This technology has matured and is now robust enough to implement machine-assistant behaviour (agents that are more independent, co-operative or capable of assisting associates).

Urlings (2003) claims that in order to compose effective human-agent teams and in order to include intelligent agents as effective members in this team, a paradigm shift in intelligent agent development is required similar to the change from the technology-driven approach to the human-centered approach in automation. He provides an example based on the operational analysis domain. He proposes that the traditional development of agent technology failed to distinguish between a software agent and a human, preventing them from being interchangeable, even though they are 'inherently different'. By establishing the difference between agents and humans, Urlings states that in a typical human-agent team both entities are not comparable but are complementary to each other by means of cooperative sharing of tasks while working in concert.

This work on first principles of human-centered automation is explained as follows: Humans are responsible for outcomes in human-agent teams; The human must therefore be in command of the human-agent team; To be in command, the human must be actively involved in the team process; To remain involved, the human must be adequately informed; The human must be informed about (able to monitor) agent behavior; The activities of the agents must therefore be predictable; The agents must also be able to monitor performance of the human; Each team member (humans and agents) must have knowledge of the intent of the other members (Urlings 2003).

We believe that human-centric agents could benefit from human cognition theories as an extension of their inherent reasoning. Researchers have demonstrated that teams can work effectively using a shared mental model, and R-CAST offers a promising technique for human-agent collaboration. A number of researchers in the multi-agent community are developing human-machine teaming systems for use in difficult and critical decision making under high workload situations. Human-machines teams are still led by humans, but we expect that human-control will be slowly transferred to machine-control as machines become autonomous and intelligent.

14.4 The Human-Centric Approach

In order to understand where human-agent collaboration fits into current agent trends, we need to have a close look at the classification of agents. We think that one such classification provides an accurate description of current agent trends. Nwana (1996) chooses to classify agent topology using categories such as mobility, reasoning, autonomy and hybrid.

Agents may have characteristics from multiple categories. For example mobile agents can posses learning attributes. Here we will focus on the third category since it is the leading area of current research in agents as well as the foundation needed for Teaming (coordination and cooperation). In this category, autonomy represents 'taking initiative' instead of simple responsive action towards the environment. Cooperation represents the 'interaction' needed to form intelligence, and the key element of intelligence is 'learning'. Nwana (1996) extends these three ideal attributes with the integration of the other categories. The resulting overlap in characteristics produces purely collaborative agents, collaborative learning agents, interface agents and ultimately smart agents. Purely collaborative agents are autonomous entities that coordinate their activities while not necessarily collaborating with other agents (proactively collaborating activities). Collaborative learning agents are self-performance improving (learning by observation) agents by observing others (agents or humans). Interface agents' typologies emphasize autonomy and learning, giving rise to application areas such as support and assistance to a user by adapting to the specific skill set so that the user 'feels' comfortable. Finally a 'smart agent' as described by Nwana (1996) should learn and interact with its external environment.

Reasoning models of agents play an important part in their existence; they have been categorized as deliberative and reactive. Purely reactive reasoning is very much like stimulus-response type, where the action is chosen based on previously defined action-response pairs. Reactive agents are most suited to less dynamic environments and for quicker response in real-time. On the other hand, deliberative reasoning is inspired from cognition theories and imitates human-like reasoning in agents. Deliberative reasoning is generally slower than reactive reasoning, but it has advantages of giving more human-like intelligence. This was one of the reasons why the early deliberative agent paradigms such as BDI became popular and widely-accepted in the agent community.

Although the BDI paradigm is widely used to mimic human intelligence, BDI agents can not be fitted in to the above definition of truly 'smart agents' since they still lack the primary ideal characteristics of 'Coordination and Learning'. We expect that one of the major steps of the next generation of agents will comprise coordination (Teaming) and, ultimately, learning.

We think that another major step in agent teaming research will be to introduce a 'human-centric' nature within an agent's architecture. The current trend in agent development is focused on its agent-only interaction, meaning that agent teaming is comprised of joint-goal operations that consist of agents as sole subordinates of the team without any human intervention. Here we distinguish between the need of a human in the loop as a colleague and as a sometimes supervisory role. This demands agent architectures to embody social ability in order to interact with the human part of the team. In Hopkins and DuBois (2005), Wooldridge describes social ability as "the ability to interact with other agents and possibly humans via some communication language."

We would like to suggest that 'interaction' with humans cannot only be via some communication language, but also can be by other means such as observation and adaptation. We would also like to suggest that truly smart agents can be complementary to a human by adopting skills similar to a human (and that may include communication, learning and coordination) rather than being a simple replacement to a human. Such a view encourages research focused on developing the agent's human-centric nature by combining one or more ideal attributes such as coordination, learning and autonomy.

14.5 Steps Toward Next Generation

The BDI agent model has the potential to be a method of choice for complex reactive systems. Future trends in agent technology can be categorized on the basis of 'Teaming' which can be divided into Multi-Agents (Teaming) and Human-Centric Agent (Human-Teaming). These two research streams have two commonalities, namely, collaboration and cooperation. Along with these, a human-centric agent possesses ideal attributes such as learning as discussed previously in the definition of a truly smart agent. Recent work on the BDI agent such as Shared Plans/Joint Intentions and JACK teams (AOS 2004) facilitates agent-only teaming. Furthermore, the addition of an ideal attribute such as learning enables agents to come closer to the goal of a human-centric smart agent.

Agent collaboration provides the opportunity for agents to share resources during their execution. Such resources are not normally available within current multi-agent system designs because resources are allocated for the use of specific teams. Team structures and team members are defined explicitly when the system is being designed. Using collaboration, agents are able to recognize when additional resources are needed and negotiate with other teams to obtain them. Collaboration is a natural way to implement human-agent teaming due the temporary and unpredictable nature of human team members.

14.6 Building a Teaming Framework

The case study presented in this section describes the proposed first steps in understanding how to implement human-agent teaming in an intelligent environment. A prototype implementation framework has been developed that allows an agent to establish collaboration with another agent or human. The framework is based on CHRIS (Sioutis and Ichalkaranje 2005), an agent reasoning and learning framework developed as an extension of JACK at the University of South Australia (Sioutis 2006). CHRIS equips a JACK agent with the ability to learn from actions that it takes within its environment. It segments the agent reasoning process into five stages based on a combination of functions extracted from Boyd's OODA loop (Hammond 2004), Rasmussen's Decision Ladder (Sioutis et al. 2003) and the BDI model (Rao and George 1995). Boyd's Orientation stage has been implemented as a collaboration module, which itself has been limited between the State and the Identification operation as shown in Fig. 14.3.

The path on the left shows the process involved in establishing a collaboration contract between two or more agents. The path on the right indicates that agents need to continuously perform assessment in order to ensure that collaboration is progressing as previously agreed upon and also whether the collaboration is yielding the required effect toward achieving each agent's

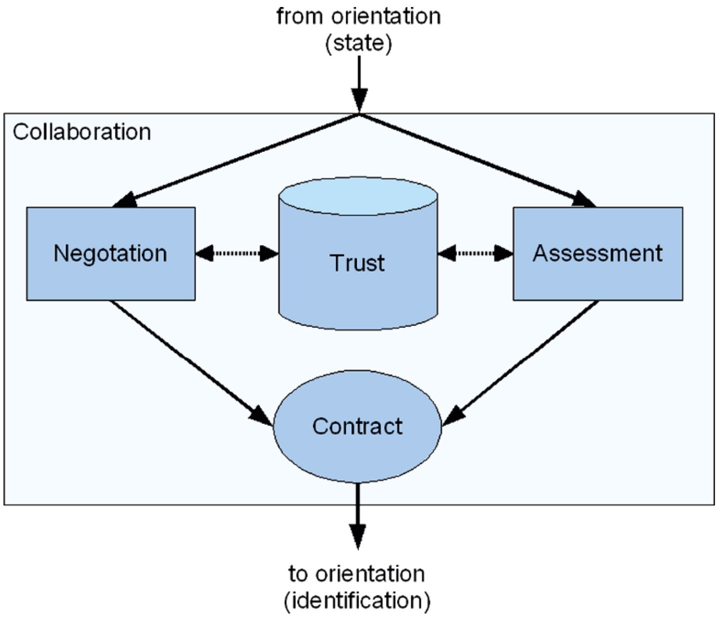


Fig. 14.3. Reasoning and collaboration

own goals. Both of these operations are highly dependent on trust, which is updated accordingly.

This implementation is based on using JACK team agents. Negotiation is performed using an authoritative Collaboration Manager Agent. Subordinate agents simply need to be able to perform the Cooperation role. The current implementation only supports goal-based collaboration relationships, where an agent negotiates for another agent to achieve a particular goal. Finally, an event called RequestCollaboration is used to ask the Collaboration Manager Agent for collaboration.

14.6.1 Decision Making Using a Human-Agent Team

A demonstration program was written that provides limited human-agent collaboration. It uses two agents. The first agent called Troop connects to a computer game called Unreal Tournament (UT) using UtJackInterface (Sioutis 2003) and controls a player within the game. The second agent is called HumanManager and is used to facilitate communication with humans encountered within the game. The program demonstrates how the Troop agent is given the goal hierarchy shown in Fig. 14.4. This Decision Making Agent is used to decide whether the entity will Defend or Attack. The Troop agent can only perform the Defend or Attack goal (mutually exclusive). This agent decides how to handle the Attack goal and then asks the Collaboration Manager Agent to organise other (friendly) human players encountered

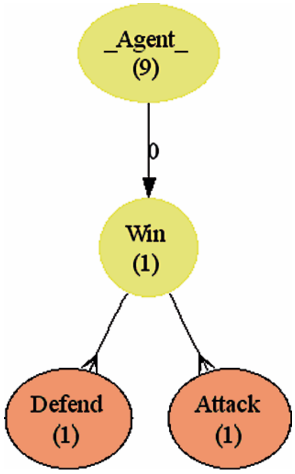


Fig. 14.4. Goal hierarchy used for demonstration

in the game to take responsibility for the alternate goal. The sequence of operations for the demonstrations is:

1. The agents Troop, Collaboration Manager Agent and HumanManager are created and a scenario.def file is used to form a team with the Cooperation role between the Collaboration Manager Agent and the HumanManager.
2. The Win goal is activated and the Defend and Attack sub-goals are subsequently activated automatically in parallel. Attack is handled by the Troop agent that subsequently attacks any enemy that comes within the field of view. For demonstration purposes, the Attack goal succeeds after the agent attacks five enemy players.
3. A RequestCollaboration message is sent to the Collaboration Manager Agent for the Defend goal. The Collaboration Manager Agent then executes an @team achieve for any sub-teams that perform the Cooperation role. The HumanManager agent then negotiates and performs assessment with the human in order to satisfy the Defend goal.

The human's point of view is acknowledged by:

(a) Asking the Human
(b) The Human Refuses
(c) The Human Accepts

The human is able to communicate with agents via text messages through UT. Figure 5a illustrates what appears on the human's monitor when a message is received from the agent. The sequence diagram shown in Fig. 5b illustrates that if the human refuses to join the team, the collaboration fails and hence both the Defend and Win goals both fail. On the other hand, the sequence diagram shown in Fig. 5b illustrates that when a human accepts to

Fig. 5a. The CMA model of a human's decision cycle (Asking the human)

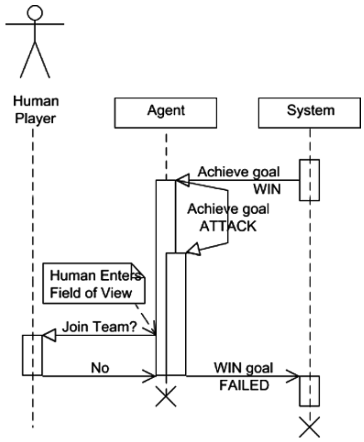


Fig. 5b. The CMA model of a human's decision cycle (The human refuses)

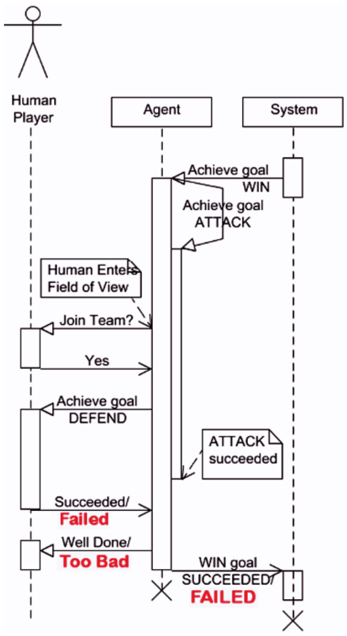


Fig. 5c. The CMA model of a human's decision cycle (The human accepts)

join the team, collaboration is formed and the human is assigned with the Defend goal. The result of the Win goal then depends on whether the human reports that he/she was successful in achieving the Defend goal.

14.7 Concluding Remarks

Intelligent agent technology is at an interesting point in its development (Valluri and Croson 2005). Commercial-strength agent applications are increasingly being developed in domains as diverse as meteorology, manufacturing, war gaming, capability assessment and UAV mission management. Furthermore, commercially-supported development environments are available and design methodologies, reference architectures and standards are beginning to appear. These are all strong indicators of a mature technology. However, the adoption of the technology is not as rapid or as pervasive as its advocates have expected. Intelligent agent technology has been promoted as the paradigm of choice for the development of complex distributed systems and as the natural progression from object-oriented programming. Is intelligent agent technology simply in need of a 'killer application' for demonstration, or are there more fundamental reasons as to why a technology that promises so much has not been more widely adopted? What does the future hold for this technology?

The development of intelligent agent applications using current generation agents is not yet routine. Certainly providing more intuitive reasoning models and better support frameworks will help, but we see behaviour acquisition as a major impediment to the widespread application of the intelligent agent paradigm. The distinguishing feature of the paradigm is that an agent can have autonomy over its execution, i.e. an intelligent agent has the ability to determine how it should respond to requests for its services. This is contrasted with the object paradigm, where there is no notion of autonomy and objects directly invoke the services that they require from other objects. Depending on the application, acquiring the behaviours necessary to achieve the required degree of autonomous operation can be a major undertaking. The problem can be likened to the knowledge acquisition bottleneck that beset the expert systems of the 1980s. Thus, there is a need for principled approaches to behaviour acquisition, particularly when agents are to be deployed in behaviour-rich applications such as enterprise management. Cognitive Work Analysis has shown promise in this regard, but further studies are required.

Alternatively, the requirement for autonomous operation can be weakened and a requirement for human interaction introduced. Rather than having purely agent-based applications, cooperative applications involving teams of agents and humans could be developed. Agent-based advisory systems can be seen as a special case of cooperative applications, but we see the interaction operating in both directions, i.e. the agent advises the human, but the human also directs and influences the reasoning processes of the agent. Existing architectures provide little in the way of support for this two-way interaction. Such interactions require that the goals and intentions of both the human and the agent are explicitly represented and accessible, as well as the beliefs that they have relating to the situation. This approach provides a convenient way to address the difficulties associated with behaviour acquisition associated with autonomous operation. By making the agent's longer term goals and intentions visible, as well as the rationale behind its immediate recommendation, this approach also provides a mechanism for building trust between humans and agents. It should also be noted that in many applications, such as cockpit automation and military decision making, full autonomy is not desirable; an agent can provide advice, but a human must actually make the decision. In these cases, we expect to see an increasing number of applications designed specifically for human teams, agent teams or a combination of both.

Learning has an important role to play in both cooperative and autonomous systems. However, the reality is that learning is extremely difficult to achieve in a general and efficient way, particularly when dealing with behaviours. The alternative is to provide the agent with predefined behaviours based on à priori knowledge of the system and modified manually from experience gained with the system. This has worked well in practice and we expect that it will remain the status quo for the immediate future.

In summary, we expect that intelligent agents will retain their architectural foundations but that the availability of more appropriate reasoning models and better design methodologies will see them increasingly used in mainstream software development. Furthermore, better support for human-agent teams will provide the impetus for the development of a new class of intelligent decision support applications (Tweedale et al. 2007).

Acknowledgements

Some of our research results presented in this chapter also appear in the proceedings of the 10th KES Conference, 2006, published by Springer and the International Journal of Decision Support Technologies, IOS Press, Volume 1, Number 4, 2007.

References

AOS (2004) JACK Intelligent Agents: Teams Manual 3.0, Agent Oriented Systems, Melbourne.

AOS (2005) What is jack? [Online, accessed 6 June 2005] http://www.agent-software.com.au/shared/products/index.html.

AOS (2006) Jack intelligent agents teams manual 4.1, [Online, accessed 2d March] http://www.agent-software.com.au/shared/resources/index.html.

Bichindaritz, I. and Marling, C. (2006) Case-based reasoning in the health sciences: What's next? Artificial Intelligence in Medicine 36, pp. 127–135.

Bratman, M. E. (1999) Intention, plans, and practical reason, center for the study of language and information.

Chan, C. (2005) An expert decision support system for monitoring and diagnosis of petroleum production and separation processes, Expert Systems with Applications 29, pp. 127–135.

Cheyer, A. and Martin, D. (2001) The open agent architecture, Journal of Autonomous Agents and Multi-Agent Systems 4(1), pp. 143–148.

Coram, R. (2002) Boyd: The Fighter Pilot Who Changed the Art of War, Little Brown and Company, Boston.

Curts, R.J. and Campbell, D.E. (2001) Avoiding information overload through the understanding of OODA loops, a cognitive hierarchy and object-oriented analysis and design, in Proceedings of the 6th International Command and Control Research and Technology Symposium (CCRTS), US Naval Academy, Annapolis, MD, pp. 312–319.

d'Inverno, M., Kinny, D., Luck, M. and Wooldridge, M. (1997) A formal specification of dMars, in Agent Theories, Architectures, and Languages, pp. 155–176.

Fan, X., Sun, S. and Yen, J. (2005a) On shared situation awareness for supporting human decision-making teams, in 2005 AAAI Spring Symposium on AI Technologies for Homeland Security, Stanford, CA, pp. 17–24.

Fan, X., Sun, S., Sun, B., Airy, G., McNeese, M. and Yen, J. (2005b) Collaborative RPD-enabled agents assisting the three-block challenge in command and control in complex and urban terrain, in 2005 Conference on Behaviour Representation in Modelling and Simulation (BRIMS), Universal City, CA, pp. 113–123.

Feigenbaum, E.A., Buchanan, B. and Lederberg, J. (1971) On generality and problem solving: a case study using DENDRAL program, in Edinburgh, B., Scotland, M. and Michie, D. eds, Machine Intelligence, Vol. 6, Edinburgh University Press, pp. 165–190.

Ferber, J., Gutkecht, O. and Michel, F. (2006) Madkit development guide, [Online accessed: 25 August 2006] http://www.madkit.org/madkit/doc/devguide/devguide.html.

Francois, F.I., Chatila, R., Alami, R. and Robert, F. (1996) PRS: A high level supervision and control language for autonomous mobile robots, in Proceedings of the IEEE International Conference on Robotics and Automation, Minneapolis, USA, Vol. 1, IEEE Press, pp. 43–49.

Frize, M., Yang, L., Walker, R. and O'Connor, A. (2005) Conceptual framework of knowledge management for ethical decision-making support in neonatal intensive care, IEEE Transactions on Information Technology in Biomedicine 9, pp. 205–215.

Group, S.D. (2005) Documentation set for swarm 2.2, [Online Accessed: 25 August 2005] http://www.swarm.org/swarmdocs-2.2/set/set.html.

Hammond, G.T. (2004) The Mind of War: John Boyd and American Security, Smithsonian Institution Press: Washington, USA.

Hanratty, T., Dumer, J., Yen, J. and Fan, X. (July 2003) Using agents with shared mental model to support objective force, in Cybernetics and Informatics (SCI 2003), 7th World Multiconference on Systemics, Orlando, USA, pp. 27–30.

Hopkins, M. and DuBois, C. (2005) New software can help people make better decisions in time-stressed situations, Science Daily pp. 1–2.

Iliadis, L. (2005) A decision support system applying an integrated fuzzy model for long-term forest fire risk estimation, Environmental Modelling and Software 20, pp. 613–621.

Jain, A., et al. (2000) Artificial Intelligence Techniques in Breast Cancer Diagnosis and Prognosis, World Scientific Publishing Company, Singapore.

Klein, G.A. (1989a) Recognition-primed decision (RPD) model of rapid decision making, IEEE Transactions on Systems, Man and Cybernetics 19, pp. 462–472.

Klein, G.A. (1989b) Recognition-primed decisions, Advances in Man-Machine System Research 5, pp. 47–92.

Klein, G.A. and Crandell, B.W. (1995) The role of simulation in problem solving and decision making.

Lange, D.B. (1997) Java Aglet Application Programming Interface (J-AAPI) White Paper - Draft 2, Technical report, IBM Tokyo Research Laboratory.

Mackworth, A. (2005) The coevolution of AI and AAAI, AI Magazine 26, 51–52.

Negnevitsky, M. (2005) Artificial Intelligence, A Guide to Intelligent Systems, Pearson Education Limited, Harlow, England.

Nwana, H.S. (1996) Software Agents: An overview, Knowledge Engineering Review 11(3), pp. 1–40.

Phillips-Wren, G. and Jain, L. eds (2005) Intelligent Decision Support Systems in Agent-Mediated Environments, IOS Press, The Netherlands.

Rao, A. and George, M. (1995) BDI Agents: from Theory to Practice, in Proceedings for the 1st International Conference on Multi-Agent Systems (ICMAS-95), AAAI Press, California, pp. 312–319.

Russel, S. and Norvig, P. (2003) Artificial Intelligence: A Modern Approach, second edn, Prentice-Hall, Inc., Englewood Cliffs, N.J. 36. Shortlife, E. (1976) MYCIN: Computer-Based Medical Consultations, Elsevier New York, NY.

Simon, H.A. (1977) The New science of Management Decision (3rd edition revised, first edition 1960), Prentice-Hall, Englewood Cliffs, NJ.

Sioutis, C. (2006) Reasoning and Learning for Intelligent Agents, PhD thesis, School of Electrical and Information Engineering, University of South Australia.

Sioutis, C. and Ichalkaranje, N. (2005) Cognitive hybrid reasoning intelligent agent system, in R. Khosla, R.J. Howlett and L.C. Jain, eds, Proceedings of the 9th International Conference on Knowledge Based Intelligent Information and Engineering Systems (KES 2005), Sep. 14–16, Melbourne, Australia, Vol. 3681(2) of Lecture Notes in Artificial Intelligence, Springer, Berlin Heidelberg New York, pp. 838–843.

Sioutis, C., Ichalkaranje, N. and Jain, L. (2003) A framework for interfacing BDI agents to a real-time simulated environment, in A. Abraham, M. Koppen and K. Franke eds, Design and Application of Hybrid Intelligent Systems, Frontiers in Artificial Intelligence and Applications, IOS Press, Amsterdam, The Netherlands, pp. 743–748.

Tweedale, J. and Cutler, P. (2006) Trust in multi-agent systems, in Proceedings of the 10th International Conference on Knowledge Based Intelligent Information and Engineering Systems (KES 2006) edited by Howlett, R.P., Gabrys, B. and Jain, L.C., Oct. 9–11, Bournemouth, England, Vol. 4252 of Lecture Notes in Computer Science, Springer Berlin Heidelberg New York, pp. 479–485.

Tweedale, J., Ichalkaranje, N., Sioutis, C., Urlings, P. and Jain, L.C. (2007b) Building a decision-making framework using agent teams, Intelligent decision technologies: An International Journal, IOS Press 1(4) (in Press.).

Urlings, P. (2003) Teaming Human and Machine, PhD thesis, School of Electrical and Information Engineering, University of South Australia.

Valluri, A. and Croson, D. (2005) Agent learning in supplier selection models, Decision Support Systems 39, pp. 219–240.

Wooldridge, M. (2002) An Introduction to Multi-Agent Systems, Wiley, Chichester.

Yen, J., Yin, J., Ioerger, T.R., Miller, M.S., Xu, D. and Volz, R.A. (2001) Cast: collaborative agents for simulating teamwork, IJCAI, pp. 1135–1144.

Editors

Gloria Phillips-Wren

Gloria Phillips-Wren is associate professor in The Sellinger School of Business and Management at Loyola College in Maryland. She previously served as Chief, Propulsion Branch, for the U.S. Army Research Laboratory focusing on future weapons systems. She was awarded a Superior Civilian Service Award and served in senior decision making positions. She is Founder and Co-Editor-in-Chief with Lakhmi Jain of *Intelligent Decision Technologies: An International Journal (IDT)* and Chair of the KES Focus Group in Intelligent Decision Technologies. She received her PhD from the University of Maryland Baltimore County and has co-published 3 books and more than 100 papers on data analysis, simulation, intelligent decision support, and their applications. She has lectured throughout the world and worked in decision support technologies with major corporations over the past 20 years. She can be reached at `gwren@loyola.edu`.

Nikhil Shripal Ichalkaranje

Nikhil Shripal Ichalkaranje is Senior Technology Advisor in the Australian Department of Broadband, Communications and the Digital Economy. He also works in the University of South Australia as adjunct Senior Research Fellow. His research interests include artificial intelligence, computer communications/networking and robotics. Nikhil holds a Masters Degree in Computer Network Systems from Swinburne University of Technology Melbourne and a PhD in Computer Systems Engineering from the University of South Australia Adelaide. Nikhil has co-edited 4 books along with several publications in the area of artificial intelligence and their applications.

Lakhmi C. Jain

Lakhmi C. Jain is a Director/Founder of the Knowledge-Based Intelligent Engineering Systems Centre, located in the University of South Australia, where he is Professor. He is a fellow of the Institution of Engineers Australia. His research interests focus on novel techniques such as knowledge-based intelligent machines, artificial neural networks, virtual systems, intelligent agents, evolutionary systems and the application of these techniques. Prior to obtaining his PhD, he worked as a Research Fellow under the Council of Scientific and Industrial Research on the design and development of educational digital computers. His publications include more than 10 co-authored books, 50 co-edited books, 15 co-edited conference proceedings, 100 research papers and 10 special issues of journals. He initiated the First International Conference on Knowledge-based Intelligent Systems in 1997. This is now an annual event. He also initiated the International Journal of Knowledge-based Intelligent Systems in 1997. He has presented a number of Keynote addresses in International Conferences on intelligent systems and their applications. He can be reached at `Lakhmi.jain@unisa.edu.au`.