Perspectives on DSS

Edited by:

John Darzentas, Jenny S.Darzentas, and Thomas Spyrou



The sixth meeting of EURO Working group on DSS Samos, Greece 20-30/6/1995

University of the Aegean

Perspectives on DSS

Perspectives on DSS

John Darzentas, Jenny S. Darzentas, and Thomas Spyrou (Eds.)

The 6th meeting of EURO Working Group on DSS held in Samos, Greece 20-30/6/1995

University of the Aegean Press

Copyright © 1996 by University of the Aegean Press.

All rights reserved. No part of this book may be reproduced or utilised in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without permission in writting from the publisher.

For information:

University of the Aegean Press 30, Voulgaroctonou Street GR 11472 Athens, Greece

ISBN 960-7475-07-0

Printed in Greece by Voulgaroctonou 28, Athens GR 11472, Greece.

Forwardii
Preface
Alphabetical list of Authors
I. Approaches and Models
DSS - Rethinking Strategic Management. <i>M. Brännback</i>
Evaluating Option-related Text Descriptions for Decision Aiding. J. Darzentas, T. Spyrou, and Ch. Tsagaris
Using a GIS as a DSS Generator. <i>P. Keenan</i>
Query-Driven Model Building In Enterprise-Wide Decision-Making Environments. <i>K.R. Lang, and A.B. Whinston</i>
A Field Service Support System Using the Computer Analysis of Networks of Queues (CAN-Q) Model. <i>H.T. Papadopoulos, and J. Darzentas</i>
Picking Your Brains: A DSS for Neurosurgery. P. L. Powell, N.A.D. Connell, P. Lees, and C.M.S. Sutcliffe
II. Applications
A Designer's Decision Aiding System: DDAS. J.S. Darzentas, T. Spyrou, E. Benaki, and J. Darzentas
Combining Techniques from Intelligent and Decision Support Systems: An Application in Network Security. <i>T. Spyrou, and R. Telesko</i>
More Effective Strategic Management with Hyperknowledge: Case Woodstrat. P. Walden, and C. Carlsson

Contents

III. Research Notes: Using DSSs

Learning Decision Making through Management Games.	
<i>T. Leino</i>	159
Putting models to work. A comparison between user- and research drive <i>K. Mossberg.</i>	n projects 167

ii

Foreword

by Tawfik Jelassi

Co-ordinator of the EURO Working Group on Decision Support Systems

The EURO working Group on DSS was launched at the EURO Summer Institute on DSS held in June 1989 in Madeira (Portugal). The participants in that institute felt it was necessary to set up a European forum for continuing exchange and collaboration on DSS-related topics. Since its foundation, the Group has met every year: in 1990 in Fontainebleau, France (host Tawfik Jelassi), in 1994 in Turku, Finland (host: Timo Leino), and in 1995 Nagycerk, Hungary (host: Andras Edelmayer) in 1993 in Sintra, Portugal (host: Antonio Rodriguez), in 1994 in Turku, Finland (host: Timo Leino) and in Samos (host: John Darzentas).

In addition to the above annual meetings, other activities of the Group have consisted of: publishing a newsletter (whose first editor was Albert Angehrn of INSEAD and the current editor is Philip Powell of the Warwick Business School); organising the first IFORS specialised conference on DSS (co-chaired by Jean Pierre Brans of the Free University of Brussels, VUB and Tawfik Jelassi of INSEAD), and editing some special issues of leading academic journals such as the European Journal of Operational Research (vol. 55, no 3, 1991 and vol. 61, nos 1-2, 1992), the Journal of Decision Systems (vol. 1, nos 2-3, 1992), and Decision Support Systems (vol. 12, nos 4-5, 1994). Another major activity that the Group is organising is the 7th EURO Mini-Conference on "Decision-Support Systems, Groupware, Multimedia and Electronic Commerce" to take place in Bruges, Belgium, from March 25-27th, 1997.

The diversity of the aforementioned activities has been, as often mentioned by the group members, truly enriching and rewarding. In my opinion, this is mainly due to the educational and cultural background of the members who come from as far afield as Scandinavia, Southern Europe, UK/Ireland, as well as Eastern and Western Europe. Furthermore, their strong motivation and involvement in the above activities have made the Group a friendly forum, cohesive and yet open to different schools of thought in the DSS field.

On behalf of my colleagues in the Working Group, I would like to warmly than our Samos Hosts, John and Jenny Darzentas as well as Thomas Spyrou and the University of the Aegean for its hospitality and for having edited the proceedings of the meeting. They undoubtedly added another brick to the further construction of our DSS network.

While we humbly take pride in our achievements to date, we look forward to having successful meetings in Ispra (Italy) in June 1996 and in Bruges (Belgium)

iii

in March 1997. We also strive for more value-added activities in the future and to increased collaboration among Group members as to further contribute to the existing body of knowledge in the DSS field.

Fontainebleau, 6 May 1996

iv

Preface

This collection of papers represents some of the work presented at the Sixth Meeting of the EURO Working Group on DSS, which took place on the island of Samos, Greece, in May of 1995.

As has been stated before, the field of DSS is rich and diverse, a claim borne out by this collection. Since the objective of the meeting was to gather together researchers who have been meeting regularly over the past five years to present their current work; to follow up on previous research; and to exchange ideas, no specific theme for the workshop was given. Hence, the resulting collection of papers do not readily form a coherent set. Rather, they are to be seen as representing various aspects of some current preoccupations in the field.

With this in mind, it is interesting to see certain trends. For instance, the domains dealt with are predominantly those with an industrial or corporate setting, although defence, spatial DSS, health units, HCI and security in networks are also present. Lacking from this year's workshop were papers on environmental DSS or group DSS. Further, the distinctions of *active* and *intelligent* DSS which in the literature of only a few years ago were 'hot topics' are not singled out for attention, but in the research presented here, it appears that such attributes are becoming standard in DSSs, and this is certainly borne out by the applications, and some papers dealing with approaches and methods.

The collection has been divided into sections. We cannot deny that these divisions are somewhat arbitrary, as for instance the emphasis of a particular application was often not just that it worked, but that it could work in other situations, whereas sometimes the analysis method was under the microscope, in other cases it was the data capture method, or the organisation of the knowledge base. Other papers straddle the theoretical /applicable boundary. Nevertheless, we make distinctions, in order for the reader to orient himself, and we believe that they will be of some use.

I. Approaches and Models

In the first part are grouped those papers which can be said to describe approaches and models. The order of papers is purely alphabetical, there is no attempt to further subgroup the papers. Brännback's paper gives a detailed and comprehensive review of Strategic Management and proposes that in spite of some opinions to the contrary, DSSs can, in fact, be very helpful here. Darzentas et al. discuss a means of evaluating option related text using a number of fuzzy operators. Some of this work has actually been implemented in a DSS described in a paper in the

Applications section. In a more technical discussion, Keenan promotes the use of

V

GIS as DSS generators and argues the case for Spatial DSSs. In their paper, Lang and Whinston attempt to deal with the very real problem that decision making incurs: namely that decision makers alternate between different perspectives and levels of details when assembling the information needed to make decisions. The authors' approach is to improve model building and make it possible to extract only the relevant parts of models needed for specific analyses, proposing a query Papadopoulos and Darzentas present a support system for driven system. maintenance services of organisations who typically have to reconcile two conflicting objectives: high level of customer service and low spares inventory. They detail the model that was used to provide the support and suggest where, with modifications, the model can be used in other areas of the service industry. Powell describes the problem and the type of analysis used to yield the results needed by a clinical unit officials needing to make decisions about allocation of resources. He concentrates on data capture, a novel use of bar-codes outside their normal stocktaking/inventory image. The success of this method has been such that the data collection will be used for other analysis leading to other types of support. As can be seen from the brief summary given above, the six papers in this section deal with approaches and models of DSS both from a theoretical but also from a very practical point of view citing actual applications.

II. Applications

In this section three papers report upon actual implemented systems, and describe their construction and reception, as well as commenting upon their special features and use. Darzentas et al. describe work on building a DSS to support users to choose which tools and techniques from a set are the most appropriate for the users' particular situation of concern. The domain here is that of Human Computer Interaction, but the methodology used to build the system is generalisable to other domains. A key feature of the system is its reliance of fuzzy logic to provide the inference machine whereby the decision aid judgements are computed. Spyrou and Telesko describe a DSS component of a wider system for intrusion detection in networks. This component's task is to provide decision support for the system's security officer who is monitoring the system and has to determine when it is appropriate to initiate countermeasures. Finally, in their paper, Walden and Carlsson describe a system that was developed for and is now in use by the forestry industry. This system supports strategic management decisions. It uses knowledge modules which are linked in a hyperknowledge fashion which makes the information more intuitively acceptable to the users.

III. Research Notes: Using DSSs

vi

In this last section, two research notes look at users of DSS and using DSSs. Leino proposes the use of Management games which simulate real life concerns of managers in order to train them in decision making and using DSSs, while Mossberg describes empirical work comparing user driven and research driven projects and reasons about which were more acceptable to and accepted by users.

In conclusion, as is to be expected with such a meeting whose aim is to present work in progress; to exchange ideas and notes with fellow colleagues and to learn from their experience, much work went on which is not captured in these papers. However, we feel it is valid and valuable to present these as an indication of the concerns and progress of workers in the field.

As organisers of the meeting we take this opportunity to publicly thank all those who participated in and made the Samos workshop the event it was, as well as the authors for their help in contributing to this volume.

John Darzentas Jenny S. Darzentas Thomas Spyrou

vii

viii

Alphabetical List of Authors

- **Benaki E.** (Research Laboratory of Samos, Dept. of Mathematics, University of the Aegean GR 83200 Karlovassi, Samos, Greece)
- Brännback M. (Institute for Advanced Management Systems Research, Åbo Akademi University, DataCity SF-20520 Åbo, Finland)
- **Carlsson C.** (Institute for Advanced Management Systems Research (IAMSR) Åbo Akademi University, DataCity A 3208, 20520 Åbo, Finland)
- **Connell N.A.D.** (Department of Accounting and Management Science, University of Southampton, Southampton SO17 1BJ)
- Darzentas J. (Research Laboratory of Samos, Dept. of Mathematics, University of the Aegean, GR 83200 Karlovassi, Samos, Greece)
- **Darzentas J.S.** (Research Laboratory of Samos, Dept. of Mathematics, University of the Aegean, GR 83200 Karlovassi, Samos, Greece)
- Keenan P. (Department of M.I.S. University College Dublin, Ireland)
- Lang K.R. (Institut für Wirtschaftsinformatik und Operations Research, Freie Universität Berlin, Berlin, Germany)
- Lees P. (Wessex Neurological Centre, Southampton Universities Hospital Trust, Southampton)
- Leino T. (Turku School of Economics and Business Administration, P.O. Box 110, 20510 Turku, Finland)
- Mossberg K. (National Defence Research Establishment, Department of Defence Analysis, S-172 90 STOCKHOLM, Sweden)
- Papadopoulos H.T. (Dept. of Mathematics, University of the Aegean, GR-832 00 Karlovasi, Samos, Greece)
- Powell P.L. (Information Systems Research Unit, Warwick Business School, University of Warwick, Coventry, CV4 7AL)
- Spyrou T. (Research Laboratory of Samos, Dept. of Mathematics, University of the Aegean, GR 83200 Karlovassi, Samos, Greece)
- Sutcliffe C.M.S. (Department of Accounting and Management Science, University of Southampton, Southampton SO17 1BJ)
- **Telesko R.** (University of Vienna, Institute for Applied Computer Science, Department of Knowledge Engineering, AT 1210 Vienna, Austria)
- **Tsagaris Ch.** (Research Laboratory of Samos, Dept. of Mathematics, University of the Aegean, GR 83200 Karlovassi, Samos, Greece)
- Walden P. (Institute for Advanced Management Systems Research (IAMSR) Åbo Akademi University, DataCity A 3208, 20520 Åbo, Finland)
- Whinston A.B. (Center for Information Systems Management, Department of Management Science and Information Systems, The University of Texas at Austin, Austin, Texas)

ix

Х

I. Approaches and Models

DSS - Rethinking Strategic Management

Malin Brännback

Institute for Advanced Management Systems Research, Åbo Akademi University, DataCity SF-20520 Åbo, Finland

In recent years new ways of strategic thinking have been proposed. Naturally the Ansoffian and Porterian approaches are still acknowledged and respected. Nevertheless, increasing competition and globalisation of markets have claimed that the approaches presented by Ansoff and Porter are insufficient or even incorrect. Certain indications that the Mintzbergian approach would be more appropriate can be found, but also attempts to combine these exist. But what are really the remedies in strategic management (SM) that have been presented, as 'schools of thought in SM', over the years? Are they really different from each other? In this discussion the role of decision support systems (DSS) is emerging again as an effective tool, despite the early doubts over their significance.

This paper is a review of early and current trends in strategic thinking and the possibilities emerging from developments in the field of DSS.

1. Introduction

The business world has over the past decade been subject to fundamental structural transition brought about by deregulations, global competition, technological discontinuities (Porter, 1990, Prahalad and Hamel 1990, 1994, Laszlo et al, 1990, Kotter, 1990, Levenhagen et al, 1993, Lorange et al, 1993). This has resulted in new customer expectations and imposed new strains on business managers. In attempts to restore competitive edge many managers are abandoning old strategy recipes and looking for new more effective guidance in turbulent environments. Yet, are the recipes really new or are they just the emperor's new clothes?

As we head toward a post-industrial society we need new concepts of the world by which to orient ourselves. Classical concepts have become unreliable and, what is worse, to some extent even irrelevant. "..*The rules of the game has changed, but the game is new*.." (Laszlo et al, 1993). In our opinion it is not so much the rules that have changed but the game is starting to resemble the adventures of Alice in Wonderland. Kanter (1989) invites corporate elephants to learn how to dance.

Strategic management is built on a search for organisational intelligence. Earlier approaches to strategic management focused on the use (or lack of use) of

analytically rational decision processes and pictured the task of intelligent management as that of failure to act rationally (March, 1988a,b, Levinthal and March, 1993). The concept of intelligent management relies on Simon's (1960) classical framework of managerial decision-making; intelligence-design-choice. Theories on strategy for exploiting comparative advantages and competitive edge were built on a conception of calculated rationality. This vision of calculated rationality continues to be the dominant thought, although it has been modified as a result of heavy criticisms of its assumptions (Minztberg, 1990a,b, 1994[a,b,c,], Ansoff, 1991, 1994).

Three basic assumptions underlie our ideas of intelligence and theories of choice: (i) pre-existence of purpose, (ii) consistency, and (iii) rationality. The focus on rational techniques make us overlook two fundamental elements of choice, i.e. intuition and tradition and faith. These can be seen as possible sources of irrationality. Why? Because they are the sources of our value systems. Decisions, which are heavily loaded with values or where the available quantitative data is insufficient tend to give leeway for values to decide the outcome of decisions.

Today the field of strategic management is engaged in hair-bending activities trying to sort out a plethora of concepts. The development seems to be a quest for classifying as much as possible and this seems to loose our touch to the practical business world and toss ourselves into the doldrums of academic argumentation. We now have Mintzberg's (1990) 'ten schools of thought', the three synthesising schools (Elfring and Volberda, 1994), the stakeholder approach (Rhenman, 1964; Rhenman and Stymne, 1965; Freeman, 1984; Näsi, 1995), the competence-based competition approach (Prahalad and Hamel 1990, 1994 a,b, Hamel and Prahalad, 1989, Hamel, 1994, Hamel and Heene, 1994, Bogner and Thomas, 1994), the Porter framework (1980, 1985) - only to mention a few. Still the classic works by Schumpeter (1934), Penrose (1959), Simon (1960), Anthony (1965), Andrews (1965), Steiner (1969) are relevant to the present business world. Nevertheless, something is obviously wrong since we are still searching for the wholly grail - this time in terms of a new paradigm (Camerer, 1985, Prahalad and Hamel, 1994b; Schendel, 1994). Given the plethora of approaches it is quite right to make additional efforts to try and unify the field of strategic management and focus its research energy (Schendel, 1994: 2 see also Rumelt et al, 1991).

In the midst of these varying number of schools from ten, synthesised (or lumped - using Mintzberg's terms) down to three schools, we propose to reduce the scope to include only one: *Strategic Management and DSS*. Before we present this approach we shall review Mintzberg' ten schools and the three synthesising schools. This will be done in section 2. In the third section we will discuss the stakeholder approach and the competence-based competition framework as these two represent further attempts to find new ways of unifying the field of SM. In the fourth section

we will outline a conceptual framework for strategic management and DSS as a tentative approach of allowing all schools to flourish.

2. Strategy as Schools of Thought

There are numerous ways of studying strategic management, some of them are more pedagogic than others. One method is to classify strategic management into **schools of thought** and this is in terms of teaching and learning an ingenious method. Then we can of course argue about the number of schools or units and sub-units to strategic management. Mintzberg (1990b) discusses ten schools, Elfring and Volberda (1994) excludes one of Mintzberg's schools (the last) and bases their three synthesising schools on nine of the ten Mintzberg's schools. Karlöf (1987) also produces a classification based on the number ten. However, the content differs widely from Minzberg's. Gilbert, Hartman, Mauriel and Freeman (1988) use six schools of thought, Näsi adds one more and arrives at seven (see Exhibit 1). Kristamuljana (1994) reduces the scope into two schools of thought.

The discussion when based on schools of thought is most likely less confusing than the attempt of describing the possible steps that should or should not be included in the strategic management process. The classic model by Ansoff (1965) must be the by far most detailed with an impressing number of 57 sub-steps. Let us here in this section restrict ourselves to Mintzberg's ten schools and focus on the tenth school, i.e. the *configurational school*. We will then briefly present a reduction of the ten schools to three schools, and we have chosen this classification since *the configurational school* is also present here. We will then return to the premises of the configurational school in section 4. We are in no way arguing that one way of thinking is more correct than the other - merely providing a collected display of what has been said so far. In addition to this school type thinking we will discuss two other 'hot' topics and we invite the reader to decide for himself/herself where these two approaches place - in terms of schools.

Mintzberg's ten schools of thought

In exhibit 2 we find the central characteristics extensively displayed in a way that makes it easy to compare the ten schools. In this section we would like to pay most attention to the configurational school since this is the school where Mintzberg places himself into. It is also the school that we will draw on in our last section when we outline the framework for a single school of thought - that of DSS and strategic management.

Karlöf's strategy schools

- 1. The experience curve
- 2. The BCG Matrix
- 3. Market attractiveness/strategic position
- 4. The Mysigma profitability graph
- 5. PIMS
- 6. Porter's generic strategy
- 7. Gap analysis
- 8. The product/market matrix
- 9. Problem detection studies
- 10. McKinsey's 7 Smodel

Näsi's strategy schools

- 1. Ansoffianism
- 2. Planning Process Approach
- 3. Portfoilio Management
- 4. Business Idea School
- 5. Porterism
- 6. Excellence and cultural Approach
- 7. Mintzbergism



Exhibit 1: Strategy as schools of thought

Mintzberg gives in a chapter in Fredrickson's book 'Perspectives on Strategic Management (pp.105-237, 1990b) an extensive discussion over the ten schools of thought in terms of premises, critique, and context and contribution, and it provides a very good starting point for anyone wishing to unravel the secrets of strategic management. More help in this pursuit can be found in Mintzberg's recent book (1994) 'The Rise and Fall of Strategic Planning', but also from the articles by Mintzberg (1990a) and Ansoff (1991) over the design school.

Let us focus on the **configurational school**, which is a summary school of thought where everything from the nine schools is allowed under certain conditions. Mintzberg provides four central premises to this school of thought. First, the behaviours of firms are best described in terms of distinct integrated clusters of dimensions concerning state and time - **configurations**. Secondly, strategy formation process is episodic (Anthony, 1965) where the form of the organisation adopts and matches with the environment engaging in certain activities for a specific time. Thirdly, the process can be that of **conceptual design**, **formal planning**, **systematic analysis**, **intuitive vision**, **individual cognition**, **collective learning**, **politics**. The driving force can be personalised leadership, culture or the external environment. The strategy can have any of the five P forms (pattern, ploy position, perspective or plan) (Mintzberg, 1987) but it must be found at its own time and context. Finally, the configurations have a tendency to sequence themselves over time.

From the premises it becomes clear that **lumping** is the key process, and here the critique comes in, '...all lumping must be considered somewhat artificial'. The configurational school attempt to **explain** by distorting - like theory tries to

simplify by distorting. Mintzberg, however, justifies his ten schools by referring to the common fact - that we prefer to think in categories - schools - in order to learn. In conclusion we would like to say that the lumping is artificial for pedagogic reasons - not for reasons that come out of a need to solve problems or make decisions, i.e. managerial reasons - **real** reasons.

Consequently, we are still left with the key problem in strategic management - **how** can future performance of an activity be improved (Brännback and Spronk, 1995), i.e. how can we provide better tools (or schools) for strategic management so that companies factually can perform better?

Underlying Dimension	Design	Planning	Positioning	Entrepre- neurial	Cognitive	Learning	Political	Cultural	Environ- mental	Configu- rational
Major Source	Selznick (1957), Andrews (1965)	Ansoff (1965)	Schendel (mid- 1970) Porter (1980)	Schumpeter(1 934), Cole, (1959), and others in economics	Simon (1945) March and Simon (1958)	Lindblom (1959,-63) Cyert and March 1963), Quinn,(1980) Weick,(1969)	Allison (1971), Pfeffer and Salanick (1978)	Rhenmann and Norman, late 1960 in Sweden; no obvious source elsewhere	Hannan and Freeman, (1977), Aldrich and Pfeffer,(1976); contingency theorists	Chandler (1962), Miller and Mintzberg, late 1970., Miles and Snow (1978)
Base Discipline	none	urban planning, systemstheory, cybernetics	economics, industrial organisation, military history	none	psychology (cognitive)	none	political science	anthropology	biology	history
Strategy	explicit perspective, unique	explicit plan, sub- strategies and programs	explicit generic positions, competitive also ploys	implicit perspective (vision), personal and unique(niche)	mental perspective	implicit patterns, often collective	ploys and positions, subunit or organi- sational	collective perspective, unique and usually, implicit	specific position	all those to the left, in context
Central actor	chief executive	planners	analyst	leader	brain	whoever can learn	whoever has power	collectivity	environ-ment	all those to the left, in context
Leadership	dominant, judgemen- tal	responsive to proce- dure	responsive to analysis	dominant, intuitive	source of cognition	responsive to initia- tives or own learning	weak, at best a player	part of collectivity	acquiesent	any, so long as categori- cal
Underlying Dimension	Design	Planning	Positioning	Entrepre- neurial	Cognitive	Learning	Political	Cultural	Environ- mental	Configu- rational
Basic Process	cerebral, simple, in- formal, judgementald eliberate prescrip-tive	formal, de- composed, staged, de- liberate, prescrip-tive	analytical, systematic, deliberate, prescrit-tive	visionary, intuitive, largely deliberate, descriptive	menatl, overwhel- ming, des- criptive	emergent, informal, messy, des- criptive	conflictive, aggressive, messy, e- mergent, deliberate, descriptive	ideological, constrained, collective, deliberate, descriptive	passive, emergent, descriptive	integrative, episodic, sequenced, all to the left, descriptive
Current and Future Status	foundation of prescription only	low, unless becomes empirical	very high	some increased interest	moderate now	growing interest	growing interest	moderate now, de-cline likely	low now, decline likely	growing interest

Exhibit 2: SCHOOLS OF THOUGHT IN STRATEGIC MANAGEMENT (adapted from Mintzberg, 1990, pp.192-197)

M. Brðnnback

Organisation	ordered, acquiscent font of given strengths and weaknesses	structured, decomposed for pro- gramming	font of competitive advantage	implicit, malleable, simple	incidental	eclectic, flexible, playful	conflicitive, disjointed, uncontrolla- ble, aggres- sive	cohesive, normative	acquiscent	any, so long as catego-rical, all to the left
Structure	machine, bureaucracy centralised formalised	machine, bureaucracy centralised formalised	machine, bureaucracy centralised formalised	simple structure, centralised, organic	not specified	adhocracy, professional bureaucracy	adhocracy, professional bureaucracy	missionary organisation	passive, bureaucratic	any, so long as configu-rational
Change	occasional quantum	periodic, incremental	piecemeal, can be ad hoc	occasional, revolutionary, opportunistic	infrequent (resisted mentally)	continual, incremental, piecemeal	frequent, piecemeal, idiosyncra-tic	infrequent (resisted ideologically)	never or rare	occasional, revolutiona-ry
Underlying Dimension	Design	Planning	Positioning	Entrepre- neurial	Cognitive	Learning	Political	Cultural	Environmenta l	Configu- rational
Environment	expedient opportu- nities, seldom threats	acquiscent, checklist of factors to be forecast or controlled	exigent in terms of es-tablished competition, analysable in economic terms	maneuvera- ble, to find a niche	overwhelm- ing for cognition	demanding, difficult	intractable, mallable	incidental	dictatorial, exigent	any, so long as categori-cal, all to the left
Intended Message	fit	formalise	analyse	envision	cope	learn	promote	coalesce	react	integrate
Realised Message	think; stratyegy making as case study	program	calculate rather than create and commit	centralise	worry (since cannot cope or invent)	play rather than pursue	hoard rather than share or produce	perpetuate rather than change	capitulate	lump rather than nuance
Vocabulary	distinctive competence competitive advantage SWOT formulation implemen- tation	program- ming, bud- geting, scheduling	generic strategy, strategic group, in-dustry and competitive analysis, portfolio	vision	map, frame, reframe, mental set, bounded rationality, cognitive style	incremeta- lism, emer- gent strategy, sense ma- king, revita- lisation	power, coalition, political games, collective strategy	myth, culture, ideology	selection, environ- mental, dynamsim, complexity, niche	configura- tional, archetype, stage, life cycle, stra-tegic revo- lution
Champions	case study proponents, rational leadership	rationalisers MBAs, professional managers	rationalisersMB As, professional managers	business press, indi- vidualists, innovators	psychologi- cally oriented	divergent thinkers, frustrated lower mgrs.	power oriented	mytholo-gists (esp. in Scand)	population ecologists	lumpers, integrators

Three synthesised schools of thought

Based on the ten schools of thought presented above it is not surprising to find statements that the field of strategic management is fragmented (Elfring and Volberda, 1994). They maintain that it is not fruitful to try and find a universal definition of strategic management, because "..the choice of a definition and application of specific strategic management techniques is greatly dependent on which paradigmatic schools of thought in strategic management one prefers..". Thus they suggest a synthesis, or lump (using Mintzbergs term's) of the accumulation of knowledge within the ten schools of thought. This results in a suggestion of three schools of thought: (i) the boundary school, (ii) the dynamic capability school, and (iii) the configurational school. Due to their character of being synthesised schools they are also based on a number of base disciplines. The reason for Elfring and Volberda suggesting a synthesising school of thought is simply the fact that when related to research and practice it is rare that only one school dominates and that a combination is more likely, i.e. in their opinion Mintzberg's classification is too fragmented.

The fragmentation appears not only in the number of base disciplines, but also (i) the classification into *descriptive* and *prescriptive*, (ii) the distinguishing between *voluntarisitc* and *deterministic*, (iii) the unit of analysis, (iv) the research area, and (v) the application of a static or dynamic perspective. The cause of fragmentation can be found in the degree of uncertainty and the virtual lack of co-ordination of research procedures and strategies between researchers (Elfring and Volberda, 1994, see also Schendel, 1994:2) and this has led to that strategic management is sometimes called 'a study in adhocracy'.

The boundary school is primarily concerned with research questions such as majority and minority participation, participation, joint ventures, network structures, i.e. make, buy or cooperate decisions (Hamel et al, 1989, Ito and Rose, 1994, Markides and Williamson, 1994).

The dynamic capability school has focused on organisational learning as means by which to develop core competencies (Argyris and Schön, 1978, Amit and Schoemaker, 1993, Levinthal and March, 1993, Nevis et al, 1995, Prahalad and Hamel, 1990, Schoemaker, 1992, 1995, Senge, 1990, Isaacs and Senge, 1992) that are hard to imitate. The dynamic capability school argues that a firm's resources and capabilities are a better basis for strategy formulation which is contrary to the traditional view of market oriented strategies (Drucker, 1974, Aaker, 1992, Day, 1990). The dynamic capability school draws on the resourced-based theory of the firm (Penrose, 1959, see also Bartlett and Goshal, 1993, Black and Boal, 1994) but also on theories on entrepreneurship (Schumpeter, 1934, see also Kanter, 1989, Stopford and Baden-Fuller, 1994).

M. Bronnback

Typical research issues of interest for this school of thought are: In fast-changing markets how do new market structures evolve? Second, what are the successful strategies associated with growth-market development? How is it that certain early players establish dominating positions before market knowledge and structures become well defined? Who has the competitive advantage, is it the first mover or the second mover? Thirdly, how do company-level activities link to group activities?

The third synthesising school is the configurational school. This school of thought is also found in Mintzberg's classification as the tenth school. Mintzberg defines the confirgurational school as a collective school for all the remaining nine schools. The key argument is that strategy is context dependent and is an episodic process where a particular type and form of organisation matches to a particular type of environment and engages in certain activities under a specific time period (Mintzberg, 1990b:182 (see also Miles and Snow, 1978). This school is occupied by finding means by which an organisation can handle change (from one configuration to another). Elfring and Volberda describes this school as having its roots in socially-oriented organisational sciences, business history, biology, and mathematical theories such as cybernetics. The configurational school is distinct from the other conceptual schools as a result of a strong empirical orientation and a systematic measurement of configurations (Elfring and Volberda, 1994, p.18).

	The Boundary School	The dynamic capabilities School	The Configurational School
Base Disciplines/ Theories	 agency theory (economics/Psycho-logy transaction costs theory industrial organisation control theories (sociology) decision-making theories (psychology) 	 resource-based theory of the firm (economics) entrepreneurship (economics) innovation theories (organisation theory) learning theories (organisational behaviour) 	 social science, history, equilibrium models (biology), catastrophe theories (maths)
Schools of thought	 positioning cognitive cultural political 	design school entrepreneurial learning environmental	 political environmental learning cognitive entrepreneurial
Problem- solving tools	 The strategy sourcing process Porter's value chain 	 The roots of competitiveness (Prahalad&Hamel) The Capability matrix (Schoemaker) 	 Archetypes (Miller&Friesen) Strategic types (Miles&Snow) FAR method (Volberda)

Exhibit 3: Synthesising schools of thought in strategic management (Elfring and Volberda, 1994)

3. Other approaches

The stakeholder approach

There are, of course, numerous other perspectives, some already fallen in oblivion and others being revived. The Stakeholder approach (Näsi, 1995) was explicitly outlined by Rhenman (1964) and Rhenman and Stymne (1965). The roots lie in the work by Barnard (1938), Cyert and March (1963), Freeman (1984) (cf. Näsi, 1995, pp. 19-32). The approach outlined by Rhenman and Stymne won appreciation in Scandinavia and came to dominate the university management teaching. Moreover the approach was used as a framework both by academics and practitioners. The dominance lasted until the dawn of the 'Porter' era, and was then tossed back into the doldrums of the plethora of theories. Once again the stakeholder approach also called stakeholder theory - came back into the limelight through Freeman's work in 1984 (Näsi, 1995, p. 20). The stakeholder approach was connected to the realm of business and society and has come to function as an umbrella, a framework, for value issues, ethics and social responsibilities of businesses. In terms of Mintzberg's ten schools of thought the stakeholder approach encompass the findings of the political, cultural and the environmental school. Contrary to Mintzberg's arguments that these three schools are on the decline (with possible exception of the political school), Näsi argues that the stakeholder has only in the 1990s seriously entered the scholarly discussion. According to Freeman (1984) the stakeholder concept may be defined as: "any group or individual who can affect or is affected by the achievements of a corporation's purpose." (Näsi, 1995, p. 21).

Typical stakeholders are owners, management, employees, customers, suppliers, lenders, government, community, media, unions, consumer groups, and environmental groups. Thus, the stakeholders are representatives of both internal and external forces of a firm. In this sense the stakeholder approach does not differ much from Porter's five forces (1980, 1985), that determine a firm's competitive strategy. Corstjens (1991) defines into the realm of *customers*, one of the five competitive forces, also *institutional customers* that are made out of various consumer interest groups (e.g. environmental groups, government, community, media, unions, etc.). Many times these are referred to as *third party* actors that implicitly or explicitly influence the competitive environment of a firm. Nevertheless, these third party actors have, according to Corstjens, influence on all five forces.

The difference between the Porter framework and the stakeholder approach lies in the treatment of the firm's *goal*. In the Porter framework the ultimate goal for a company is to create a sustainable competitive advantage, whereas Rhenman and Stymne (1965) quite promptly maintains that *the firm itself quite simply has no goals!* With reference to today's business environment this conclusion appears highly dubious, if not somewhat dangerous. The goals, it appears, are more or less

expressed in terms of contributions and rewards (stakes and pay-off) and then we have not actually come very far from the Porter framework where rewards indeed are the proceeds that stem out of a sustainable competitive advantage. The stakeholder approach is displayed in figure 1 below.



Figure 1: The stakeholder approach (Näsi, 1995)

Much in the same line of reasoning lies the *competence-based competition* framework by Prahalad and Hamel (1990, 1994 a,b, Hamel and Prahalad, 1989, Hamel, 1994, Hamel and Heene, 1994, Bogner and Thomas, 1994).

This framework, as presented by Bogner and Thomas (1994, p. 114), relies on internal forces which is based in the company's core competence and on sustainable competitive advantage (SCA) which is the external trait (Fig.2). Bogner and Thomas argue that a core competence cannot be a core competence unless it does not give a firm a competitive advantage in a given market place by satisfying a customer need better than a competitor. It is important to make this distinction for avoiding both a misallocation of resources to activities that *do not* lead to SCA and an under-allocation of resources to those activities that *could* lead to a SCA.



Figure 2: Competence-based competition (Bogner and Thomas, 1994)

Core competencies are internal traits; skills and understanding acquired over time - a company's *knowledge-base. Competitive advantage* is the external trait, the competitive edge a company has on a market based on a bundle of goods and/or services offered at a price charged. Core competencies are quite often not even interesting for the customer, i.e. in the drug industry; a core skill in refining a specific drug. Such a thing as competitive advantage out of *pure luck* is an artefact according to Bogner and Thomas. Core competencies are unique and ad hoc competitive advantage lack the trait of skill and replicability.

With reference to this competence-based framework we can quite easily continue to develop it towards the realm of decision support systems (recently also called knowledge-based support systems (KSS)). It soon becomes obvious that decision support systems in this framework can make significant contributions to strategic management. Decision support systems will become an inherent element in the concept of *core competence*.

In the next section we will continue to develop this line of reasoning and show how DSS indeed can make significant contributions to strategic management. We will also discuss the relevance of 'schools of thought' in strategic management - whether ten or three, and show how a DSS enables the combination all ten schools and how strategic planning will become strategy formation as in the premises of the **configurational school**.

4. Strategic Management with Decision Support Systems

Past doubts

The possible contributions of computers to the field of strategic management has long been doubted. In 1965 Anthony stated (p.45) "..*it is because of the varied and unpredictable nature of the data required for strategic planning that an attempt to design an all-purpose, internal information system is probably hopeless. For the same reason, the dream of some computer specialists of a gigantic data bank, from which planners can obtain all the information they wish by pressing some buttons, is probably no more than dreams." Simon (1960) was also doubtful but saw some possibilities. Both authors acknowledged the fact that many companies were trying to develop some kind of support systems for decision-making. Ackoff (1967) was heavily criticising the premises for management information systems (MIS) and Little (1970) described the problems that managers faced when working with models. Little consequently outlined what a model should <i>be, do and not do,* and this is relevant still this day. On mathematical models and their possible contribution of computers in this context (p.60 and p.62):

"Even the most sophisticated model is unlikely to include enough data, arranged in the right way to provide an automatic answer to questions that may be asked of it. The dream of a giant computer that gives instant answers to such questions (...) is only a dream at present, and is likely to remain so for a long time. Not many companies have such models. The task of preparing one and keeping it current is expensive both in staff time and executive time (...) no one trusts the validity of the assumptions incorporated in the model. Nevertheless, the number of companies that are attempting to construct such models seems to be growing rapidly."

"In the management control process, the sophisticated model is probably of extremely limited usefulness..the manager operates within policies already established; he does not explore the implications of new policies..A few computer enthusiasts foresee an era of 'automatic management', when computers will make all the decisions now made by operating managers (...) we believe such a prediction to be highly unrealistic."

It is true that part of the explanation to why computer-based support systems have had so little success in supporting managerial decision-making lie in the realm of measurement, i.e. the handling of quantitative issues (measurable) and qualitative issues (hard to measure). Although most decisions taken in firms have an underlying financial structure where results are expressed in monetary units, but from this does not follow that money is the *only* basis of measurement, or even that it is the most important basis (Anthony, 1965). There are such measures as market share, productivity measures, enrolment, etc. that are very useful too, not to mention nonquantitative expressions such as quality, ability, cooperation, and so on. Or how do we model **values** and **value contradictions** (cf. Brännback and Malaska, 1995) Some successful work has been done in quantifying value judgements (Bana e Costa and Vansnick, 1994). These recent developments towards solving most of the problems of yesteryear (concerning modelling hard and soft data) have been possible because of the fast developments in the field of information systems technology.

Yet, the criticism of decision support systems (DSSs) having little or no significant impact on decision-making (Alter, 1981, 1992, Angehrn and Jelassi, 1993, Bell, 1992, Keen, 1987, Stabell, 1987) has led to the fact that DSSs in strategic management is still a rarity (Carlsson, 1991, Turban, 1991, Holtham, 1992), and many people still believe that DSSs cannot provide support in strategic management. We strongly believe that DSS can provide support for strategic management (Brännback, 1994, Carlsson and Walden, 1994). The past limitations on the use of empirical data do not exist with modern information technology.

Managers limited **knowledge of** and **ability to** use computers are less relevant to the modern business world as an increasing number of managers can use computers today. On discussing information handling within strategic planning, management control, and operational control Anthony stresses the degree of detail and the cost of providing information. The expertise of the specialist in management control systems does not need to know *about* the information handling, but in a general way *the capabilities and limitations* of computing equipment. How to make the best combination of the available equipment or how to construct the best program are questions and tasks for engineers. Anthony cites McFarlan (pp.96-97) on what the tasks of each party are and should be, and that they need to cooperate in order to provide results with any significant impact on performance:

"The information handling specialist has a dynamic responsibility to utilize new techniques for the improvement of information available (...) Operating people bring a bias, which favors preservation of the status quo. The information handling analyst (...) his bias is towards the introduction of too much change (...) While it is not his responsibility to decide what information the manager *should* have, it is his responsibility to show the manager information he *can* have (...) Successful implementation of an improvement in an information handling system requires coordination efforts by both information handling specialists and operating personnel."

Prospects

In the fast changing business environment of today, managing of the firm's knowledge base and to what extent it matches the changing competitive conditions becomes critical (the second premise of the configurational school). This can be seen as managing the distinctive knowledge base over time, containing both technical and social knowledge.

A simple model of how strategic information is processed may be presented as follows:

 $Data \rightarrow Information \rightarrow Knowledge \rightarrow Expertise$

Data must be enacted by the company before it can be considered as information relevant to the company. Next, information need to be integrated, assembled in a meaningful way, thus yielding knowledge. Knowledge itself must then be transformed to lead to expertise. Expertise is an articulated set of complementary complex visions. Visions may occasionally be conflicting but they provide a unique ability to a wide variety of problem solving (Nonaka, 1991, 1994, Prietula and Simon, 1989, Durand, 1993).

The developments of information systems technology has resulted in new opportunities for developing DSS **that have an significant impact in strategic management**. The DSS will make strategic management a process resembling very much the premises of the configurational school where the strategy making process will be that of conceptual design, formal planning, systematic analysis, but also intuitive vision, individual cognition, collective learning, and politics. Furthermore the DSSs will be context specific, designed for **that** company and **that** business, where the form and the content of a DSS will adopt and match to the environment engaging in certain activities for a specific time. The reason for this lies simply in the characteristics of **strategic management** which we define as a *process for creating a sustainable competitive advantage* (Brännback 1993).

These systems are visual, in that they display numerical data and graphical displays together, they model qualitative data and quantitative data, and the user can easily orient him/herself in the system (Angehrn and Lüthi, 1990, Angehrn, 1991a, b, Brännback, 1994, Walden et al, 1995). The systems are genuine decision **support** systems in that it is expected that the user will make the actual decisions. The system only supports the decision-maker by helping him/her to distinguish the relevant information from the trivial.

In developing these kinds of DSSs, that also can be named **knowledge-based support systems** (KSS), the process has to focus on the user and his/her needs. This is in no way taken care by conducting a few interviews, it requires a iteration process that can be very tedious and conflicting, but so very rewarding once the KSS has been completed. It requires teaching the users to not only use the KSS but also to deal with maintenance and updating. In other words we are looking for the type of commitment that Prahalad and Hamel (1989, 1990) describe concerning a firm's core competence and strategic intent. The KSS will also support a **dynamic** strategy making process, not only because it is thought to be used frequently, but because it will make the user focus on the key issues in strategic management thus resulting in the system being constantly verified and validated. We will have **active** decision support (Keen, 1987, Jelassi et al, 1987, Angehrn, 1991a,b). There are, of course, those that will warn you from activating the user too much by introducing end-user modelling (Gass, 1990). The KSS should also support a company's efforts in becoming an active learning organisation, which increasingly is seen as one of

the keys to a firm's future success (cf. Senge, 1990, Isaacs and Senge, 1992, March and Levinthal, 1993).

Changes in the business environment will no-longer make the KSS useless, provided that these options have been taken into account already in the development process, i.e. in choosing development tools and in constructing the system. A shift from rule-based systems to object-oriented can be seen, yet the rules will probably never disappear.

We can refer to at least one examples of a successful implementations of KSS -Woodstrat - that has been developed through iteration, that is modular, that has made the strategy making process indeed very dynamic, and that is being used (Carlsson and Walden, 1994, Walden et al, 1995, Carlsson and Walden, 1996). Woodstrat has been developed at the Institute for Advanced Management Systems Research (IAMSR) since 1992, thus the process is indeed tedious. In 1993 the author was involved in developing a prototype (MOCK - Many Options for Complex Knowledge) for a Finnish drug company (Brännback, 1993, 1994) however the company was very small and the strategy making process was not seen (by the company) as that complex as requiring any specific KSS, yet the elements for complexity did exist. Woodstrat, again, has been developed for the forest industry which in terms of size and volume makes the strategic decisions very complex. Therefore the company involved could easily see the benefits from implementing a KSS in their strategic management process.

Although these are only some scarce examples they are convincing in showing how modern KSS (DSS) should be developed and implemented and especially Woodstrat shows that KSS can have a significant impact in strategic decision-making. Even if MOCK never passed the prototype phase it was not rejected for reasons based on the characteristics and functionalities of the system, but because the company had not made a careful potentiality study, i.e. where they prepared to commit themselves in a way needed to guarantee a successful development and implementation process. The commitment both in terms of human and financial resources was insufficient.

Another very significant outcome of these two examples is that they have shown that the academic discussion is fruitless as such, it acquires flesh only after it has found support in empirical work (recall Elfring's and Volberda's description of the configurational school). It is also highly disputable whether carrying on an endless description and redescription of strategic management processes in terms of schools of thought will help companies finds means by which they can effectively solve their pressing problems. The real key to contributing to the theoretical and practical findings in strategic management is to work in close cooperation with companies. The development of KSSs is one good way, and here the iteration process is vital, because it will build the commitment on both sides. It will provide the field of strategic management with new insights on managerial work and it will widen the use of KSS thus significantly contributing to the field of decision support systems. The important contribution to the field of DSS also lies in the orientation towards the fields of **organisational behaviour** and **cognitive science** that are very relevant to the field (Angehrn and Jelassi, 1993, Brännback, 1993) but that have previously to a very large extent been overlooked.

5. Conclusions

In this paper we have been reviewing the strategic management literature of 'the old days' and recent years. We have described a field that has long been searching for the holy grail and always seemed to have fallen short. We present different schools of thought which has been one way of trying to find the missing link in strategic management. We have shown fragmented classifications and lumped classifications, we have also presented some other ways of looking at strategic management, i.e. stakeholder approach and the competence-based competition approach. Finally we have then discussed the role of computers in this context and specifically that of DSS or KSS.

We have argued that many of the past problems with DSS and their failure came from technological short-comings, not so much from that the ideas and visions concerning DSS were bad. We have then discussed recent experiences with DSS with some examples. Our conclusion is that we do not need ten, seven, three or any other number of schools in strategic management, but that the configurational school serves as a meaningful basis for building DSS that will have an impact on decision-making in strategic management. What is more it will bring the fields of strategic management and DSS closer together. The strength lie in that with DSS strategic management will become heavily anchored in empirical data, and this is bound to bring new insight into the theories of strategic management.

Assessing the environment is the starting point in the process of competition. Assessing the environment is not like acquiring knowledge and judgement as simple as in a technical exercise. Rather it has to be regarded as a company becoming an open learning system.

References

- 1. Aaker, D. A.: Strategic Market Management 3rd ed., John Wiley & Sons, New York, 1992
- Ackoff, R. L.: Management Misinformation Systems, Management Science, Vol. 14, No. 4, pp. B-147-156, 1967

- Alter, S.: Transforming DSS jargon into principles for DSS success, DSS-81 Transactions, First International Conference on Decision Support Systems, pp. 8-27, 1981
- 4. Alter, S.: Why persist with DSS when the real issue is improving decision making?, in Jelassi, T., Klein, M. R., Mayon-White, W. M. (Eds.) IFIP Transactions, DSS: Experiences and Expectations, North-Holland, 1992
- 5. Amit, R., Schoemaker, P. J. H.: Strategic assets and Organizational Rent, Strategic Management Journal, Vol 14, pp. 33-47, 1993
- 6. Andrews, K. R.: The Concept of Corporate Strategy, Dow Jones-Irwin, Inc. Homewood, Ill., 1971
- Angehrn, A. A., Lüthi, H-J.: Intelligent Decision Support Systems: A Visual Interactive Approach, Interfaces, Vol. 20, No. 6 (November-December), pp. 17-28, 1990
- Angehrn, A. A.: Modeling by Example: A link between users, models and methods in DSS, European Journal of Operational Research, Vol. 55, pp. 296-308, 1991a
- 9. Angehrn, A. A.: Designing Humanized Systems for Multiple Criteria Decision Making, Human Systems Managment, Vol. 10, pp. 221-231, 1991b
- Angehrn A. A. and Jelassi T., "DSS Research and Practice in Perspective", paper presented at the 4th Meeting of EURO Working Group on Decision Support, Sintra Portugal, 8-11 July, 1993
- 11. Ansoff, I.: Corporate Strategy, McGraw-Hill Book Company, New York, 1965
- 12. Ansoff, I.: Critique of Henry Mintzberg's The Design School: Reconsidering the Basic Premises of Strategic Management, Strategic Management Journal, Vol. 12, pp. 449-461, 1991
- Ansoff, I.: Comment on Henry Mintzberg's Rethinking Strategic Planning, Long Range Planning, Vol. 27, No. 3, pp. 31-32, 1994
- 14. Anthony, R. N.: Planning and Control Systems, A Framework For Analysis, Harvard University, Boston, 1965
- 15. Argyris, C., Schön, D.: Organizational Learning, Reading, Massachusetts, 1978
- Bana e Costa, C. A., Vansnick, J-C. 1994a. MACBETH An Interactive Path Towards the Construction of Cardinal Value Functions, International Transactions In Operational Research, Vol. 1, No. 4, pp. 489-500
- 17. Barnard, C. I.: The Functions of the Executives, Harvard University Press, Cambridge, Mass. 1938
- 18. Bartlett, C. A., Goshal, S.: Beyond The M-form: Toward a managerial Theory of the Firm, Strategic Management Journal, Vol. 14, pp. 23-46, 1993

- 19. Bell P. C., 1992 : "DSS: Past, Present and Prospects", Journal of Decision Systems, Vol. 1, No. 2-3, pp. 127-137
- 20. Bell, 1992
- Black, J. A., Boal, K. B.: Strategic Resources: Traits, Configurations and Paths to Sustainable Competitive Advantage, Strategic Management Journal, Vol. 15, 131-148, 1994
- Bogner, W. C., Thomas, H.: Core Competence and Competitive Advantage: A Model and Illustrative Evidence from the Pharmaceutical Industry, in Hamel, G., Heene, A. (Eds.) Competence Based Competition, John Wiley & Sons, Chichester, pp. 111-143, 1994
- 23. Brännback, M. Effective Strategic Market Management With Knowledge-Based Support Systems, Ser. A: 411, Åbo Akademy Press, 1993
- 24. Brännback, M. Decision Support Systems for Strategic Management, Journal of Decision Systems, Vol. 3, No. 2, 1994
- 25. Brännback, M., Malaska, P. Cognitive Mapping Approach Analyzing Societal Decision-Making, World Futures, Vol. 44, pp. 1-15, forthcoming
- Brännback, M. Spronk, J.: A Multidimensional framework for Strategic Decisions, Paper to be presented at the XIIth International Conference on MCDM, 19-23.6. 1995 Hagen, Germany
- 27. Camerer, C.: Redirecting research in Business Policy and Strategy, Strategic Management Journal, Vol. 6, pp. 1-15, 1985
- Carlsson, C., Walden, P.: Strategic Management and Hyperknowledge: Re-Engineering Strategic Planning in the Forest Industry, Proceedings of the 2nd SISnet Conference, Barcelona, 1994
- 29. Carlsson, C., Walden, P.: Cognitive Maps and a Hyperknowledge Support Systems in Strategic Management, Group Decisions and Negotiations, *forthcoming*
- 30. Carlsson, C.: New Instruments for Management Research, Human Systems Management, Vol. 10, No. 3, pp. 203-220, 1991
- 31. Corstjens M., 1991 : Marketing strategy in the Pharmaceutical Industry, Chapman&Hall, London
- 32. Cyert, R. M., March, J. G. A Behavioral Theory of the Firm, Prentice Hall, Engelwood Cliffs, N. J., 1963
- 33. Day, G.: Market Driven Strategy Process for Creating Value, The Free Press, New York, 1990
- 34. Drucker, P. F.: Management Tasks, Responsibilities, Practices, William Heinemann Ltd, London, 1974

- 35. Durand, T.: The Dynamics of Cognitive Technological Maps, in Lorange, P., Chakravarthy, B., Roos, J., Van De Ven, A. (Eds.), Implementing Strategic Processes, Change, Learning & Cooperation, Blackwell Business, Oxford, pp. 165-189, 1993
- 36. Elfring, T., Volberda, H. W.: Schools of Thought in Strategic Management: Fragmentation, Integration or Synthesis, Paper presented at the EIASM Workshop on Schools of Thought in Strategic Management 12-13.12. 1994
- 37. Freeman, R. E.: Strategic Management a Stakeholder Approach, Marchfield, MA, Pitman Publishing, 1984
- 38. Gass, S. I.: Model World: Danger, Beware of the User as Modeler, Interfaces, Vol. 20, No.3, pp. 60-64, 1990
- 39. Gilbert, D. R., Hartman, E., Mauriel, J. J., Freeman, R. E.: A Logic for Strategy, Ballinger, Cambridge, 1988
- 40. Hamel, G., Prahalad, C. K.: Strategic Intent, Harvard Business Review, May-June, pp. 63-76, 1989
- 41. Hamel, G., Doz, Y.L., Prahalad, C. K.: Collaborate with Your Competitors and Win, Harvard Business Review, Jan-Feb., pp. 133-139, 1989
- 42. Hamel, G.: The Concept of Core Competence, in Hamel, G., Heene, A. (Eds.) Competence Based Competition, John Wiley & Sons, Chichester, pp. 11-33, 1994
- 43. Hamel, G., Heene, A.: Competence Based Competition, John Wiley & Sons, Chichester, 1994
- 44. Holtham C., 1992 : "Architectures for executive support systems towards a prototype top manager workstation", in IFIP Transactions, DSS: Experiences and Expectations, Jelassi T., Klein M. R., Mayon-White W. M. (Eds.), North Holland, pp. 275-290
- 45. Isaacs, W., Senge, P.. Overcoming limits to learning in computer-based learning environments, European Journal of Operational Research, pp. 183-196, 1992
- 46. Ito, K., Rose, E. L.: The Genealogical Structure of Japanese Firms: Parent-Subsidiary Relationships, Strategic Management Journal, Vol. 15, pp. 35-51, 1994
- Jelassi T., Williams K., Fidler C. S., 1987: "The Emerging Role of DSS: From Passive to Active", Decision Support Systems: The Internatioanl Journal, Vol.3, No. 3, pp. 299-307
- 48. Jelassi, T.: Gaining Business Value From Information Technology: The Case of Otis Elevator, France, European Management Journal, Vol. 11, No.1, pp. 62-73, 1993
- 49. Kanter, R. M.: When Giants Learn to Dance, Simon & Shuster, New York, 1989
- 50. Karlöf, B.: Business Startegy in Practice, John Wiley, Chichester, 1987
- 51. Keen, P. G. W.: Decision Support Systems: The Next Decade, Decision Support Systems, Vol. 3, No. 3, pp. 253-265, 1987
- 52. Kotter, J. P.: What Leaders Really Do, Harvard Business Review, May-June, pp. 103-111, 1990
- 53. Kristamuljana, S.: Flexible Strategies: Two Schools of Thought, Paper presented at the EIASM Workshop on Schools of Thought in Strategic Management 12-13.12. 1994
- 54. Laszlo, E., Masulli, I., Artigiani, R., Csányi, V.: The Evolution of Cognitive Maps, New Paradigms for the Twenty-First Century, Gordon and Breach Science Publishers, Amsterdam, 1993
- 55. Levenhagen, M., Porac, J. F., Thomas, H.: The Formation of Emergent Markets, in Lorange, P., Chakravarthy, B., Roos, J., Van De Ven, A. (Eds.), Implementing Strategic Processes, Change, Learning & Cooperation, Blackwell Business, Oxford, pp. 145-164, 1993
- 56. Levinthal, D. A., March J. G.: The Myopia of Learning, Strategic Management Journal, Vol. 14, pp. 95-112, 1993
- 57. Little J. D. C., 1970 : "Models and Managers: The Concept of a Decision Calculus", Management Science, Vol. 16, No. 8 (April), pp. B-466-485
- Lorange, P., Chakravarthy, B., Roos, J., Van De Ven, A.: Implementing Strategic Processes, Change, Learning & Cooperation, Blackwell Business, Oxford, 1993
- March, J. G.: Bounded Rationality, Ambiguity, and the Engineering of Choice, in Decision making, descriptive, normative, and prescriptice interactions, Bell, D. E., Raiffa, H., Tversky, A. (Eds.), Cambridge University Press, 1988a
- 60. March, J. G. 1988b. The Technology of Foolishness, In Decisions and Organizations, March, J. G. (Ed.), Basil Blackwell Ltd., Oxford, pp. 253-265
- 61. Markides, C. C., Williamson, P.J.: Related Diversification, Core Competence and Corporate Performance, Strategic Management Journal, Vol. 15, 149-165, 1994
- 62. Miles, R. E. Snow, C. C.: Organizational Strategy, Structure and Process, McGraw-Hill, New York, 1978
- 63. Mintzberg, Henry The Strategy Concept I: Five Ps For Strategy, California Management Review, Vol 30, No.1, pp.11-24, 1987a
- 64. Mintzberg, H.: The Design School: Reconsidering The Basic Premises of Strategic Management, Strategic Management Journal, Vol. 11, 171-195, 1990a

- Mintzberg, H.: Strategy Formation Schools of Thought, In Fredrickson, J. W. (Ed.), Perspectives On Strategic Management, Harper Business, New York, pp. 105-236, 1990b
- 66. Mintzberg, H.: The Rise and Fall of Strategic Planning, Prentice Hall, New York, 1994c
- 67. Näsi, J.: Understanding Stakeholder Thinking, Gummerus Kirjapaino, Jyväskylä, 1995
- 68. Näsi. J.: Strategic Thinking as Doctrine, Development of Focus Areas and New Insights, In Näsi, J. (Ed.) Arenas of Strategic Thinking, Foundation for Economic Education, Helsinki, 1991
- 69. Nevis, E. C., DiBella, A. J., Gould, J. M.: Understanding Organizations as Learning Systems, Sloan Management Review, Winter, pp. 73-85, 1995
- 70. Nonaka, I.: The Knowledge-Creating Company, Harvard Business Review, November-December, pp. 96-104, 1991
- 71. Nonaka, I.: A dynamic theory of organizational Knowledge Creation, Organization Science, Vol. 5, No.1, pp. 14-37, 1994
- 72. Penrose, E. T., The Theory of Growth of the Firm, Basil Blackwell, New York, 1959
- 73. Porter, M. E.; Competitive Strategy, The Free Press, New York, 1980
- 74. Porter, M. E.: Competitive Advantage, Creating and Sustaining Superior Performance, The Free Press, New York, 1985
- 75. Porter, M. E.: The Competitive Advantage of Nations, MacMillan Press Ltd, London, 1990
- Prahalad, C. K., Hamel, G.: The Core Competence of the Corporation, Harvard Business Review, May-June, pp. 79-91, 1990
- 77. Prahalad, C. K., Hamel, G.: Competeing for the Future, Harvard Business School Press, Boston, 1994a
- Prahalad, C. K., Hamel, G.: Strategy as a Field of Study: Why Search for a New Paradigm, Strategic Management Journal, Vol. 15, Special Issue (Summer), pp. 5-16, 1994b
- 79. Prietula, M. J., Simon H. A.: The Expert in Your Midst, Harvard Business Review, Jan-Feb. pp. 120-124, 1989
- 80. Rhenman, E., Stymne, B.: Företagsledning i en förenderlig värld. Stockholm, Aldus/Bonniers, 1965
- 81. Rhenman, E.: Företagsdemokrati och Företagsorganisation. Stockholm, Thule, 1964
- 82. Rumelt, R. P., Schendel, D., Teece, D. J.: Strategic Management and Economics, Strategic Management Journal, Vol. 12, pp. 5-29, 1991

- Schendel, D., Introduction to the Summer 1994 Special Issue 'Strategy: Search for New Paradigms', Strategic Management Journal, Vol. 15, pp. 1-4, 1994
- 84. Schoemaker, P. J. H., How to Link Strategic Vision to Vore Capabilities, Sloan Management Review, Vol. 34, No. 1, pp. 67-81, 1992
- 85. Schoemaker, P. J. H.: Scenario Planning: A Tool for Strategic Thinking, Sloan Management Review, Winter, pp. 25-40, 1995
- 86. Schumpeter, J. A.: The Theory of Economic Development, Harvard Business press, Cambridge, 1934
- 87. Senge, P. M.: The Fifth Discipline, Currency and Doubleday, New York, 1990
- Simon., H. A.: The New Science of Management Decisions, Harper&Row, New York, 1960
- 89. Stabell, C. B.: Decision Support Systems: Alternative Perspectives and Schools, Decision Support Systems, Vol. 3, No. 3, pp. 243-251, 1987
- 90. Steiner, G. A.: Top Management Planning, The Macmillan Company, London, 1969
- 91. Stopford, J. M., Baden-Fuller, C. W.: creating Corporate Entretreneurship, Strategic Management Journal, Vol. 15, pp. 521-536, 1994
- 92. Turban E., 1990 : Decision Support and Expert Systems: Management Support Systems, Macmillan Publishing Company, New York
- 93. Walden, P. Kokkonen, O. Carlsson, C.: Woodstrat: A Support System for Strategic Management *forthcoming* in Turban, McLean and Wheterbe (Eds.) Introduction to Information Technology

DSS - Rethinking Strategic Management

Evaluating Option-related Text Descriptions for Decision Aiding

J. Darzentas, T. Spyrou, Ch. Tsagaris.

Research Laboratory of Samos Department of Mathematics University of the Aegean

This short paper presents an approach for evaluating the meaning of text descriptions regarding the appropriateness of a method, a tool, or, in general, of an approach, for solving ill-structured, ill-defined problems. This evaluation approach is based on the framework of fuzzy sets and in particular, test score semantics [9]. The text descriptions are descriptions of problems which correspond to a number of (sub)problems belonging to human activity systems elicited and represented through the use of soft systems methodology (SSM)[1,2,6]. A number of papers [3,4] have described the overall approach through specific problem spaces. An implemented decision support system to support computer system designers in the area of human computer interaction, which was based on the above approach is also described in a previous paper. The specific theme of this paper is to present and discuss some further aspects of the reasoning of such a system.

The system of relevant activity subsystems is the main vehicle for providing a representation of the problem space useful for the purpose of aiding the decision maker in his decision making as to which approach, tool etc. to use to tackle his problem.

This system is defined here as the space which consists of activity subsystems S_j , and their relationships as follows:[4]

$$\begin{split} & [S_j, \, {\textstyle{\textstyle{\mathsf{R}}}}_{{\scriptscriptstyle{\mathsf{S}}}}^{\mathsf{mt}_i}, \, \Diamond, \, \mathsf{mt}_i, \, \mathsf{x}] \\ & [S_j, \, {\textstyle{\textstyle{\mathsf{R}}}}_{{\scriptscriptstyle{\mathsf{S}}}{\scriptscriptstyle{\mathsf{S}}}{\scriptscriptstyle{\mathsf{k}}}}^{\mathsf{mt}_i}, \, S_k, \, \mathsf{mt}_i, \, \mathsf{x}] \end{split}$$

where, S_j is the activity subsystem j, j = 1..N, mt_i is the tool i, i = 1..k, $R_s^{mt_i}$ is the relationship identified within S_j which could also be in relation to a tool, in this case i. (\diamond denotes that the relationship is actually an attribute of S_j which stems out of the properties of mt_i).

 $R_{SS_k}^{mt_i}$ is the relationship between S_j , S_k again possibly in relation to mt_i , e.g. corresponding to pre and post-condition. Finally x is an empirical measure of how much $R_s^{mt_i}$ is satisfied by mt_i .

The relevant activity subsystems S_j are elicited using Soft Systems Methodology (SSM) [1,2].

The resulting attributes associated with subsystems (subproblems) selected by the designer, can be separated into groups according to the tools they are associated to. Each of these groups can now be evaluated in order to provide recommendation as to which tools are more appropriate to be used for the particular problem. The evaluation of these groups of attributes is carried out with the aid of test score semantics and is described in the next section.

Use of test score semantics and fuzzy sets.

The relationships R are usually expressed in text form and are considered as a collection of fuzzy constraints. That is, a number of propositions constituting the meaning of the relationship between the subsystem Sj and the tools in terms of relevancy of those tools identified and S_i .

Assume that the user requesting decision aid has settled with a set of subsystems as being relevant to his concern at the current stage of problem tackling. The real decision problem will be to evaluate the meaning of the usefulness of each tool which, via the relationships to the subsystems selected by the user, will appear suitable. Following the test score semantics procedure to evaluate each relationship (fuzzy constraint) the user will provide a score ts_i for each relationship, which will describe the degree to which the relationship is satisfied. Constraint satisfaction means how much the values of the linguistic variables implied in the proposition representing the user's concern satisfy the relationship (fuzzy constraint).

According to this approach his test scores assigned to every relationship will give overall test scores for the groups of attributes discussed elsewhere [3], which correspond to each tool. The highest of these overall test scores may be taken as a very good indication that the corresponding tool is the most appropriate currently.

However it must be noted that the suggested approach is an attempt to evaluate the meaning of relationships in terms of a proposition expressing concern. In other words it is an attempt to identify the most "meaningful" action to be taken by the designer in terms of using a tool to proceed with solving his problem. In that context it is worth mentioning that Zadeh [9] suggests that the overall score by itself does not represent the meaning of the proposition of concern, but one has to consider the actual process leading to that score. As a result the overall scores here

cannot always reflect the appropriateness of a tool over another in relation to a design situation. For example a tool may be moderately appropriate but it may satisfy (moderately) a large number of links (fuzzy constraints), while another one may strongly satisfy one or two constraints only. The fact that only a few constraints are very much satisfied, may be enough to overpower the case of the great number of constraints moderately satisfied in a fuzzy environment.

$[A, \mathbf{R}_{A_1}^{mt_1}, \diamond, mt_1, x]$	[ts ₁]	a bit
$[A, \mathbf{R}_{A_2}^{mt_1}, \Diamond, mt_1, x]$	[ts ₂]	a quite
$[B, \mathbf{R}_{B_1}^{mt_1}, \diamond, mt_1, x]$	[ts ₃]	so & so
$[A, \mathbf{R}_{AB_1}^{mt_1}, B, mt_1, x]$	[ts ₄]	substantially

The table above gives as an example some partial scores which, instead of being crisp numbers selected within a range, could also be expressed linguistically through fuzzy quantifiers, or they could be expressed as fuzzy numbers. As a consequence, the partial scores are also fuzzy sets with corresponding membership functions.

The aggregation of test scores is the key to exploiting the degrees of freedom of expression offered by fuzziness. So far a number of aggregation operators [7,8] have been tried for evaluating the meaning of a problem description, that is aggregation of the value (score) of a constraint of satisfaction given by an expert with the value (score) given by the user (usually of the degree of importance to him of a problem description) as well as the aggregation of the scores corresponding to the subproblem associated to each tool. The recommendation given by the system is based on that particular score.

A number of experiments are planned to identify the most appropriate aggregation operators for the purpose. These experiments will be based on subjects who will be asked to evaluate individual subsystems as well as combinations of them from a number of domains. It is expected that, as has been found by others [11] not all specific operators are appropriate for all cases but that combinations of them depending on a number of operators.

Also in the quest for the appropriate operators to support efficient fuzzy reasoning mechanism combinations of fuzzy rules are currently used in the following fashion.

In a general case of problem description by a user [3] for which when R is the generic relationship "satisfaction" the overall score (evaluation) of the user's problem in relation to specific tools is calculated as follows:

Every subproblem (activity subsystem) is associated with a generic fuzzy rule of the type:

IF	<pre>{satisfaction of tool's potential to tackle subproblem}</pre>	AND	{relevancy of the subproblem to users' overall problem}	THEN	{appropriateness oftool}
----	--------------------------------------------------------------------	-----	------------------------------------------------------------------	------	-----------------------------

In the following figure this generic rule is represented via the membership functions \mathbf{s}_i , \mathbf{r}_i , and \mathbf{a}_i . A user may evaluate satisfaction and relevancy with fuzzy quantifies etc. with corresponding membership functions the \mathbf{s}_i , and \mathbf{r}_i' . The appropriateness, represented via the membership function \mathbf{a}_i' , is then calculated in the same fashion as in fuzzy control [5]. For satisfaction and relevancy Zadeh's sup-min operator [9] is used, while Mamdani's minimum operation rule is used for calculating the appropriateness [5] i.e. the THEN. For the AND the minimum operator is used.

A user selects a number of subproblems from those spanning the overall problem space. Also each subproblem's rule corresponds possibly to more than one tool. Hence for the selected subproblems the rules are applied for each tool separately to calculate its appropriateness. The rules are combined via the also operator i.e. the union. Other operators might possibly be used.



References

- [1] CHECKLAND, P.B., Systems Thinking, Systems Practice. Wiley, 1981.
- [2] CHECKLAND, P.B., SCHOLES, Soft Systems Methodology in action. Wiley, 1990.
- [3] DARZENTAS, J., DARZENTAS, J.S., SPYROU, T., Towards a Design Decision Aiding System: (D/DAS). Amodeus Project Document: TA/WP9, 1993.
- [4] DARZENTAS, J., DARZENTAS, J.S., SPYROU, T., Defining the Design "Decision Space": Rich Pictures and Relevant Subsystems. Amodeus Project Document: TW/WP 21, 1994.

- [5] LEE, C.C., Fuzzy Logic in Control Systems, IEEE Trans. on Systems, Man and Cybernetics, SMC, 20, (2), pp. 404-435, 1990.
- [6] LEWIS, P.J., Rich Picture Building in the Soft Systems Methodology. *European Journal of Information Systems*, Vol 1, no.5, 1992, pp. 551-360.
- [7] MIZUMOTO M., Pictorial Representations of Fuzzy Connectives, Part I: Cases of T-norms, T-conorms and Averaging Operators, Fuzzy Set and Systems, 31, pp.217-242, 1989.
- [8] MIZUMOTO M., Pictorial Representations of Fuzzy Connectives, Part II: Cases of Compensatory Operators and Self-Dual Operators, Fuzzy Set and Systems, 32, pp.45-79, 1989.
- [9] ZADEH, L.A., Knowledge Representation in Fuzzy Logic, IEEE Transactions on Knowledge and Data Engng, 1, (1), pp.89-100, 1989.
- [10] ZADEH, L.A., KACPRZYK, J.,(eds), Fuzzy Logic for the Management of Uncertainty, Wiley, 1992.
- [11] ZIMMERMANN H.-J., ZYSNO P., Latent Connectives in Human Decision Making, Fuzzy Set and Systems, 4, pp.37-51, 1980.

Using a GIS as a DSS Generator

Peter Keenan

Department of M.I.S. University College Dublin, Ireland.

The continuing development of DSS applications requires that new technologies be exploited to allow new classes of decision be supported. This paper discusses the use of a Geographic Information System (GIS) as a Decision Support System (DSS) generator to create Spatial Decision Support Systems (SDSS). Many important areas of DSS application, such as routing or marketing, make use of spatial information. This paper argues that the development of such systems will allow effective support be provided for decisions which make use of spatial data.

Keywords : Decision Support Systems, Geographic Information Systems.

Introduction

The concept of Decision Support Systems(DSS) is generally regarded as having originated with the work of Gorry and Scott-Morton (1971). While there are many definitions of a DSS, there is general agreement that these systems focus on decisions and on supporting rather than replacing the user's decision making process. There is also a general consensus in the definitions of DSS that both database and model component are usually required to fully support decisions.

In the period since the early 1970s DSS has emerged as an important component of information systems, with an increasing research output by DSS researchers. The growth is reflected in literature surveys of DSS applications research (Eom and Lee 1990). This growth in the importance of DSS has taken place against a background of rapidly changing computer technology. The introduction of widely available personal computers and their one hundred fold increase in performance has facilitated the development of a wide range of DSS applications. Other new technologies such as multimedia or the use of CD-ROM storage open up possibilities for decision support applications which could not have been easily implemented in the early years of DSS.

One area of information systems that has expanded enormously in recent years is that of Geographical Information Systems (GIS). As is the case with DSS there are numerous definitions of GIS; for a review of these see Maguire (1991). The majority of these definitions describe a system for storing and displaying spatially or geographically related data. GIS has its origins in the fragmented use of computer technology in the 1960s for automated cartography and address matching software. The development of comprehensive GIS software required improvements in graphics and database techniques. By the 1980s a number of different forms of commercial GIS software became available, including widely used products such as ARC/INFO[™]. These systems generally were used on UNIX workstations. At the end of the 1980s, PC based GIS software become available, reflecting the increase in PC performance to levels previously associated with workstations.

By the 1990s many different types of commercial GIS software were on the market and the technology had achieved widespread use in its traditional areas of application, for example in forestry and natural resource applications. The increasing use of GIS was both facilitated by, and responsible for, the increasing volume of digital spatial data becoming available in developed countries.

Geographic Information Systems

A GIS makes use of geographical and attribute data. Attribute data, addresses, populations, etc., is associated with geographical data. Geographical data may be represented as points, lines or polygons. Attribute data can be handled easily using a conventional database management system (DBMS). It is the handling of the geographical data, such as the existence of rivers, roads or contour lines that requires the use of the special techniques that characterise the use of GIS. A GIS, as distinct from a mapping program, will have a database of geographic data, allowing linkages between different types of data and the ability to query this spatial data. For example a GIS database query might allow identification off all roads with a certain distance of a river. Therefore, while traditional database approaches can support queries on the attribute data, GIS is defined by its ability to cater for spatial queries.

The growth of GIS has been driven by the importance of spatially related data. It is estimated that up to 80% of data needed for the activities of business and government is spatially related (Franklin 1992). The growth in GIS use also reflects the decreased cost of the technology. This explosion in the use of computer technology can also be seen in other areas, where a vicious cycle of declining hardware costs leads to larger software sales and therefore reduced software costs. This trend has lead to some mapping software becoming available on a mass market basis, for example the inclusion of mapping facilities in the Lotus 1-2-3 Release 5 spreadsheet. This mass market use of mapping and GIS products creates a large demand for spatial data, increasing amounts of which are becoming available. Decision makers who make use of basic mapping products, such as those provided with Lotus 123, are likely to become aware of the need for more sophisticated software. Recent improvements in mainstream PC technologies

P. Keenan

facilitate this increase in the use of spatial data. These include inexpensive gigabyte sized hard disks, large high resolution colour monitors, graphics accelerators and CD-ROM storage.

Many areas of DSS application are concerned with geographic data, including one influential early example of a DSS, the GADS system (Grace 1976). A more recent important prototype DSS, Tolomeo (Angehrn and Lüthi, 1990) uses a geographical context for the development of visual interactive techniques. However there has been limited impact by mainstream GIS techniques on DSS research. This situation is beginning to change. Recent DSS textbooks are including GIS as a component of management support systems (Mallach 1994, Turban 1995). While these texts stress the usefulness of geographically related information, they do not provide a complete picture of the relationship of GIS to other management support systems. GIS related research is beginning to make an appearance at conferences associated with DSS. For example a recent paper by Crossland and Wynne (1994) presented empirical evidence of the usefulness of a spatial approach to decision making. This paper was presented at the Hawaii International Conference on System Sciences, a conference associated with DSS rather than GIS based applications.

GIS techniques are beginning to have an impact on DSS applications. The survey by Eom, Lee and Kim (1993) identified marketing and routing as important areas of DSS application, both of these fields are recognised as areas of GIS application (Maguire 1991). In the area of routing Bodin, who was identified by Eom, Lee and Kim (1993) as an important author in routing DSS, has argued for incorporation of GIS in routing (Bodin and Levy 1994). Keenan (1995) proposed a classification of routing problems with respect to their spatial content and the usefulness of a SDSS. A number of GIS products are aimed at marketing applications, for example the Tactician GIS.

Within the field of GIS there are many who consider GIS software to provide decision support. Indeed as Maguire (1991) points out, some authors have argued that a GIS is a DSS. A substantial number of GIS based applications are described as being DSS. A recent GIS conference was entitled "DSS 2000". This view of GIS as a DSS is not entirely without support in the existing definitions of DSS. Alter (1980) proposed an influential framework for DSS which includes data driven DSSs that do not have a substantial model component. Standard GIS software could be regarded as an *analysis information system* in Alter's framework, the critical component of such a system being the database component.

However, in many cases, the description of these GIS applications as being DSS is not based on reference to the DSS literature. This may be a reflection of the trend identified by Keen (1986) for the use of any computer system, by people who make decisions, to be defined as DSS. Even where GIS contains the information relevant to a decision, they are usually general purpose systems, not focused on a particular decision. There are many problem areas where GIS techniques can make an important contribution but where models are needed to fully support the decision. For these areas at least, a GIS cannot be said to be a DSS as such a system lacks the support that the use of models can provide.

Spatial Decision Making

SDSS can therefore be seen as an important subset of DSS, whose potential for rapid growth has been facilitated by technical developments. The availability of appropriate inexpensive technology for manipulating spatial data enables SDSS applications be created. The benefits of using GIS based systems for decision making are increasingly recognised. Muller (1993) identified SDSSs as a growth area in the application of GIS technology. However the value of SDSS is not determined by its innovative use of technology. Instead the contribution of these applications will be determined by the need for a spatial component in decision making.

I suggest that three categories of decision maker may find that SDSS can make a contribution to their decisions. The first group is in the traditional areas of application of GIS, in disciplines such as geology, forestry, and land planning. In these fields GIS was initially used as a means of speeding up the processing of spatial data, for the completion of activities which contribute directly to productivity. In this context the automated production of maps, in these disciplines, has a role similar to that of data processing in business. In these subject areas there will be growth of decision making applications in much the same way as data processing applications evolved into DSS in traditional business applications. The greater complexity of spatial information processing and its greater demands on information technology have lead to the ten to fifteen year time lag identified by Densham (1991).

The second group of decision makers for whom SDSS can make an important contribution is in fields such as routing or location analysis. Although the spatial component of such decisions is clear, DSS design has in the past been driven predominantly by the management science models used. In the future these models will be incorporated into GIS based SDSS, providing superior interface and database components to work with the models. This synthesis of management science and GIS techniques will provide more effective decision making, as Keenan(1995) has argued in the context of vehicle routing.

The third group of decision makers who will find SDSS important include those where the importance of spatial data is somewhat neglected at present. In disciplines such as marketing, additional possibilities for analysis are provided by the availability of increasing amounts of spatially correlated information, for example demographic data. Furthermore the availability of geographically convenient product supply locations relative to customers is an important tool of market driven competition. In these areas the availability of user friendly SDSS to manipulate this data will lead to additional decision possibilities being examined which are difficult to evaluate without the use of such technology (Grimshaw 1994).

Building a DSS using a GIS as a generator

Because of the variety of decision making situations where spatial information is of importance, it is clear that SDSS will be an increasingly important subset of DSS in the future. It is useful to examine the relationship of GIS software to such systems. Densham (1991) discusses the development of DSS in the context of the framework proposed by Sprague (1980). In Sprague's framework a DSS may be built from tools, individual software components that can be combined to form a DSS. At a higher level in Sprague's framework are DSS generators, from which a specific DSS can be quickly built. Sprague envisioned that different specific DSS applications would require different combinations of the generator and tools. Sprague used GADS (Grace 1976), which can be regarded as a form of GIS, as an example of a DSS generator.

In building DSS, specific generators have been designed for certain classes of problem. In other situations general purpose software such as spreadsheets or DBMS packages have been regarded as generators. In modern DBMS and spreadsheet software, the use of macro and programming languages facilitates the creation of specific applications. Various generators have strengths and weaknesses in terms of their provision of the key components of a DSS; an interface, a database, and models. In the case of a spreadsheet, modelling is the basic function of the software; various interface features such as graphs are provided, but database organisation is simplistic. DBMS software, such as Access or Paradox, has good database support, provision for interface design through the use of forms, report and charts, but almost no modelling support. In this case the modelling support has to be added to the specific DSS built from such a system.

The decision regarding the appropriate mix of DSS tools and the use of a generator is an important component of the process of building a DSS. However there is a very real sense in which the types of DSS design considered for a given class of problem are a function of the available DSS generators for that class of problem. In practice a small DSS project could be built, using an off-the-shelf spreadsheet or DBMS package, in less time than it would take to fully evaluate the full range of alternative methods of constructing the DSS. Therefore the DSS solutions actually constructed are strongly influenced by the perceived availability of suitable generators. Therefore the effective application of DSS technology can benefit from additional generator software becoming available. Awareness of the potential of the use of GIS based systems as DSS generators will lead to problems, currently being approached in other ways, being approached by using a SDSS.

There is evidence that GIS software is becoming increasingly suitable for use as a generator for a SDSS. As GIS designers gain a greater awareness of decision making possibilities, their systems will be designed to facilitate interaction with models. GIS software provides a sophisticated interface for spatial information. Even limited functionality GIS software will provide the ability to zoom and to display or highlight different features. GIS provides database support that is designed to provide for the effective storage of spatial data. Furthermore GIS software provides a link between the interface and database to allow the user to easily query spatial data. However in terms of the widely accepted definition of a DSS, a GIS is not a complete DSS because of the almost complete absence of models or support for the organisation of models.

The construction of a specific DSS from GIS software is possible however, by incorporating models that make use of the GIS database and interface. In this context low end GIS and desktop mapping products may prove more manageable for applications design than full workstation based GIS systems. While these desktop systems lack the power of a full GIS, they may be able to make effective use of data which has been prepared for a specific purpose using a full feature GIS.

However some developments in GIS software since 1990 may make possible the use of standard software as the basis for an SDSS. An example of this type of software is the ArcView package from ESRI. As its name suggests, this software is primarily designed as to allow the user view and query spatial data. ArcView is available for the Windows, Macintosh and UNIX environments. It is intended that the full ARC/INFO package will be required for some GIS operations. ArcView has its own macro language; Avenue, the ability to interact with SQL database servers, and the ability to use platform specific links with other software. Together with its ability to support spatial queries, these characteristics make ArcView a potential generator for many types of SDSS software.

The incorporation in many GIS products of macro languages, such as Avenue in ArcView or Mapbasic in Mapinfo, facilitates their use to construct a DSS. In other cases GIS software allows the use of external procedures. Such linkages may not be entirely integrated, but nevertheless allow the useful combination of GIS software and models contained in external programs. An example of such software is found in Jankowski (1995), who discusses the integration of GIS software and multiple choice decision making (MCDM) techniques in a DSS. Routesmart (Bodin and Levy, 1994) provides vehicle routing functionality within the TransCad GIS.

The use of GIS as a DSS generator can make use of new facilities for interaction between software, techniques such as object linking (OLE), dynamic data

exchange, and open database connectivity (ODBC). These techniques will allow data pass from the GIS to modelling software which can provide facilities not found in the GIS itself. Present software development trends suggest an object oriented future, in which small specialised applications, or applets, will be available for use as part of a larger package. In the Windows environment the development tools such as Microsoft Visual Basic or Borland Delphi. In this context the DSS generator, the GIS, will provide the main interface and database facilities, with applets used for additional modelling or interface requirements.

Conclusion

Given the advances in computer technology in general and GIS techniques in particular, I suggest that a growing subset of DSS applications in future will be those built using a GIS as a DSS generator. This class of DSS will make an important contribution, not because of its use of the latest technology, but because it will allow decision makers incorporate a spatial dimension in their decision making. This spatial dimension, which is not fully catered for in traditional DSS designs, is an important feature of many areas of DSS application. These potential areas of application including fields, such as routing or marketing, which have been important fields of DSS application in the past. The challenge for DSS builders is to achieve an appropriate synthesis of modelling techniques and interface and database approaches, drawn from the GIS domain, to provide effective decision support for these areas.

References

- Alter, S., 1980 : Decision Support Systems: Current Practice and Continuing Challenges, Addison-Wesley, Reading, USA.
- Angehrn, A. A., and Lüthi, H-J., 1990 : "Intelligent Decision Support Systems: A Visual Interactive Approach", *Interfaces*, 20, 6, 17-28.
- Armstrong, A. P., and Densham, P. J., 1990 : "Database organization strategies for spatial decision support systems", *International Journal of Geographical Information Systems*, 4, 1, 3-20.
- Bodin, L., and Levy, L., 1994 : "Visualization in Vehicle Routing and Scheduling Problems", *ORSA Journal on Computing*, 6, 3, Summer 1994, 261-268.
- Crossland, M.,D.; Wynne, B.,E., 1994 : "Measuring and testing the effectiveness of a spatial decision support system", Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences. Vol.IV: Information Systems: Collaboration Technology, Organizational Systems and Technology Edited by Nunamaker, J.,F., Sprague, R.,H., IEEE Comput. Soc. Press, 1994.

- Densham, P.,J., 1991 : "Spatial Decision Support Systems", *Geographical Information Systems, Volume 1 : Principles*, edited by Maguire, D.J., Goodchild, M.F. and Rhind, D.W., Longman, 403-412.
- Eom, H., and Lee; S., 1990 : "Decision support systems applications research: A bibliography (1971-88)", *European Journal of Operational Research*, 46, 333-342.
- Eom, S., Lee, S., and Kim, J., 1993 : "The intellectual structure of Decision Support Systems (1971-1989)", *Decision Support Systems*, 10, 19-35.
- Franklin, C., 1992 : "An Introduction to Geographic Information Systems: Linking Maps to Databases", *Database*, April 1992, 13-21.
- Gorry, A., and Scott-Morton, M, 1971 : "A Framework for Information Systems", *Sloan Management Review* 13, Fall 1971, 56-79.
- Grace, B. F., 1976 : "Training Users of a Decision Support System", *IBM Research Report RJ1790*, IBM Thomas J. Watson Research Laboratory, 31 May 1976.
- Grimshaw, D.J. 1994 : Bringing Geographical Information Systems in Business, Longman, 100-111
- Jankowski, P., 1995 : "Integrating geographical information systems and multiple criteria decision-making methods", *International Journal of Geographical Information Systems*, May-June 1995, 9, 3, 251-73.
- Keen, P., 1986 : "Decision Support Systems: The Next Decade", *Decision Support Systems: a decade in persepective*, edited by McLean, E and Sol, H.G., North-Holland.
- Keenan, P., 1995 : Spatial Decision Support Systems for Vehicle Routing, Working Paper MIS 95/10, Graduate School of Business, University College Dublin.
- Maguire, D.J., 1991 : "An Overview and definition of GIS", *Geographical Information Systems, Volume 1 : Principles*, edited by Maguire, D.J., Goodchild, M.F. and Rhind, D.W., Longman, 9-20.
- Mallach, E.G., 1994 : Understanding Decision Support Systems and Expert Systems, Irwin., 428-435.
- Muller, J-C., 1993 : "Latest developments in GIS/LIS", International Journal of Geographical Information Systems, 7, 4, 293-303.
- Sprague, R., 1980 : "A Framework for the development of Decision Support Systems", *MIS Quarterly*, 4/4, December 1980.
- Turban, E., 1995 : Decision Support and Expert Systems 4th ed., Prentice-Hall International, 241-242.

Query-Driven Model Building In Enterprise-Wide Decision-Making Environments

Karl R. Lang* and Andrew B. Whinston**

*Institut für Wirtschaftsinformatik und Operations Research Freie Universität Berlin Berlin, Germany;

**Center for Information Systems Management Department of Management Science and Information Systems The University of Texas at Austin Austin, Texas.

1. Introduction

In order to respond to new challenges in an increasingly complex and dynamic environment, modern management is using a vast amount of knowledge from various sources. Depending on the particular problem being investigated, managers switch between different perspectives and levels of detail when searching for the relevant pieces of knowledge required to provide an appropriate answer. However, lacking a centralized knowledge management facility, individual managers' access to knowledge is restricted to a relatively small subset of the collective organizational knowledge, depending on their status and function within the organization. This may inhibit the recognition of the interactions and interdependencies relevant to the problem under study.

Conceptually, we are looking for an enterprise modeling system $(EMS)^1$ which automatically builds and executes task-specific models as needed in response to queries posed by the user. The focus of this paper is on how we can improve model building and how we can extract the relevant parts of models to support specific analyses of enterprise-wide corporate issues. We discuss and propose

¹ We use the term enterprise modeling in accordance with the definition provided in Petrie (1992), p.19, where enterprise is defined as "a collection of business entities ... in functional symbiosis," and thus differs from the usage in the business re-engineering area. Business entities mean organizational (sub)units and (groups of) people and functional symbiosis refers to the interactions among a set of intraorganizational as well as interorganizational entities sharing a common goal. Hence, the scope of enterprise modeling explicitly includes external partnerships like relationships of an organization with its suppliers, subcontractors, customers, and the public. Some authors, for example Carter et al (1992) on p.4, use the term organizational decision support system to describe concepts similar to our notion of enterprise modeling.

ideas which we see as promising steps towards accomplishing this difficult endeavor. There are two, essentially disjoint, research efforts, one based in the artificial intelligence community and the other in the decision support systems community, which study model building and reasoning with multiple models. This paper draws upon both of these efforts and develops a synergistic framework for future enterprise modeling systems.

We envision a system whose reasoning about a particular organization is based upon a library of model components (or model fragments) representing significant organizational phenomena from different perspectives and at different levels of detail. Accomplishing this requires access to multiple sets of heterogeneous model fragments which differ in several dimensions, some of which might even be mutually inconsistent. We need to address the issue of model representation and model organization. That is, we need a language for expressing relationships of different kinds and for expressing underlying assumptions controlling and guiding their applicability.

Researchers in the DSS and Artificial Intelligence (AI) communities have proposed several frameworks which provide partial solutions to this formidable problem. Model management in the DSS field can be seen as a natural extension of previous work in management science and operations research. It has advanced mathematical modeling from a state where modeling was an uncoordinated task, whose success depended mainly on the technical skills and expertise of the user, to a state where systems actually know about certain types of mathematical models and appropriate solvers [Geoffrion (1987), Liang (1988), Mannino et al (1990), Krishnan and Westernberg (1991), Muhanna and Pick (1992), Basu and Blanning (1993), Dolk and Kottemann (1993), Raghunathan et al (1993)]. The emphasis of AI research has been put more on the issue of the explicit representation of modeling assumptions, and the usage and exploration of qualitative knowledge, and less on model integration and in particular on solver integration [De Kleer and Brown (1984), Kuipers (1988), Addanki et al (1991), Falkenhainer and Forbus (FF) (1991), Rickel and Porter (1992), and Weld (1992)].

2. A Framework for Query-Driven Enterprise-Wide Modeling Systems

We propose a model building strategy which builds models as needed in response to user queries. Given a query, the model formulation problem can be defined as selecting the relevant model fragments and generating a composite, task-specific model which is coherent and useful in answering it. Using different sets of assumptions and various kinds of knowledge ranging from general, qualitative knowledge to specific and precise numerical models, managers analyze organizational questions from different perspectives and at different levels of detail. Given a particular task, model building is guided by the selection of an appropriate perspective and level of detail, a modeling decision for which little support is found in current decision support system technology. When modeling a certain organizational phenomenon, it is crucial to focus on the relevant aspects of the situation under investigation, that is, to include all the relevant objects and constraints, but also to exclude irrelevant ones and ignore unnecessary details. We suggest the software architecture depicted in figure 1 for designing such an EMS.



Figure 1: EMS Software Architecture.

The EMS is designed as an interactive software tool which supports decision making and problem solving when exploring various business scenarios. It comprises five functional modules: The query manager, the model manager, the candidate evaluation module, the solver, and the report generator. The query manager provides the interface between the EMS and the user, typically an organizational decision maker or a technical assistant to one. It processes user's queries such as, "How does an increase in price affect net income?" and translates them into a set of executable statements which are submitted to the model manager.

The core of the EMS is the model manager which controls access to models and data in the organizational knowledge base. The enterprise modeling framework requires first the building of a general-purpose organizational knowledge base that describes a variety of organizational objects, activities, and processes. The domain theory is represented as a library of model fragments, each describing an independent aspect from a particular viewpoint. It contains general organizational laws and rules as well as relationships that are very specific to a particular company. The organizational knowledge described in the domain theory could be obtained from research results in the organizational behavior field, which tries to formulate theories about organizations in general, that is, to find relationships that help understand the behavior of a wide variety of organizations. Since those relationships are supposed to hold for any particular organization of the class, they tend to be very qualitative in nature. Organization-specific information, on the other hand, is derived from historical data and experience accumulated within a particular company, and therefore, tends to be much more precise. This information is often encoded in a quantitative, management science/operations research (MS/OR) type of model like optimization, simulation, or forecasting models. The explicit representation of modeling assumptions in terms of abstraction level, approximation, perspective, level of detail, and granularity is another essential feature in enterprise modeling. Reasoning about those assumptions enables the EMS to identify a suitable collection of compatible model fragments and to build consistent, composite models in response to a query. Typically, there is no unique composite model and the model manager might find several feasible models, called candidate models, and passes each of them on to the next EMS module.

The candidate evaluation module then collects all candidate models and chooses the best candidate as the final scenario model. In this context, best means the simplest possible model that is coherent, comprehensive, and appropriate for the task. The solver module selects the adequate solution method and then solves or simulates the scenario model chosen by candidate evaluation. Finally, a report generator is employed as a post processor in order to translate the model solution, that is, the output of the solver, into an intelligible answer which can be presented to the user in return to the original question.

3. The Organizational Knowledge Base

In this section, we discuss the organizing principles of the organizational knowledge base (OKB) underlying our enterprise modeling framework. We view the OKB as a repository of organizational knowledge whose purpose is to provide a resource of sharable and reusable model pieces for helping to better understand, explain, and predict organizational phenomena in a variety of different situations. In order to achieve the necessary depth and versatility, the OKB needs to contain knowledge of different types: (i) relationships among organizational variables encoded as quantitative or qualitative constraints; (ii) their preconditions and associated modeling assumptions that define the presuppositions under which they hold; and (iii) knowledge about knowledge expressed as metarules which relate modeling assumptions to each other. The observation that a model consists of more than just a set of relationships, because a model always assumes a particular modeling context, leads us to a definition of an EMS model component (or model fragment) where the modeling assumptions are explicitly and separately expressed from the actual relationships. Each model fragment has two sections, one contains the specification of modeling assumptions (conditions section) and the other (relations section) contains the actual constraints and relationships that apply if the model assumptions hold. Model fragments are essentially of the form

fragment <NAME> (input port) (output port)

{verbal description of the functionality of the model fragment}

conditions

precondition-specifications

relations

relationship-specifications

end

where <NAME> is an identifier of a particular model fragment instance, input port is a list of the variables whose values need to be provided, either by computing them in other model fragments or by importing them as exogenous quantities, output port is a list of the variables which are computed by this model fragment, and which can be shared with other fragments. The conditions section contains precondition-specifications, which define the modeling assumptions that an instantiation of a model fragment depends on. Lastly, the relations section contains relationship specifications, which would be constraints of a particular modeling language. We only assume that internally, that is, within a single model fragment, the relationships are of a homogeneous type. Across model fragments, heterogeneous relationship specifications are permitted by using several modeling languages.

Before a model composition algorithm can actually search the model base and identify task-specific, relevant model fragments, it needs sufficient information to be able to evaluate the predicates in the model assumption section. This extra information needs to be either derived directly from the query or inferred from meta knowledge present in the OKB. Meta knowledge is to be specified separately from the model fragments as a set of rules. These meta rules express integrity constraints which rule out incoherent and inconsistent combinations of modeling assumptions, and also imply additional conditions as a consequence of modeling assumptions that have been already established.

Now, on the next couple of pages, we give an example that illustrates how the EMS principles discussed above apply to the development of an OKB model. For our purpose, showing just a small segment of an OKB shall be sufficient to demonstrate the essential features of an OKB. The complete enterprise description would obviously be much more elaborate. For the sake of simplicity, we have left out some details, and included some further restrictions in a separate rules section. In particular, rule R-1 limits the model base to quasi-static models, rule R-2 selects QSIM as the only solver for purely qualitative scenario models, and rule R-3 chooses RCR as the only solver for semi-qualitative models².

The intelligence of the EMS, however, resides mainly in the interaction graph, shown in figure 2, which represents knowledge about organizational knowledge. It is used in the OKB as a comprehensive hypermodel of the enterprise. More specifically, the interaction graph relates organizational variables, organizational relationships, modeling assumptions, and model fragments to each other. The nodes of the interaction graph represent organizational variables and arcs connecting two nodes indicate the existence of a relationship between the two corresponding variables. Arc labels identify model fragments containing such relationships. The specification of a relationship cannot be directly obtained from the interaction graph, but must be retrieved from the relations section of the containing model fragment. Likewise, modeling assumptions are to be found in the conditions section of the identified model fragment. Finally, self-loops, that is, arcs which leave from and return to the same node, indicate that the corresponding variable could be treated as being exogenous.

In the lower left corner of figure 2, we can see, for example, that model fragment f2 contains a relationship among the variables usage of information technology (IT)

 $^{^2}$ QSIM and RCR are two modeling languages which allow the user to represent purely qualitative and semiqualitative information; see Hinkkanen et al (1995) for a detailed discussion.

and Productivity (Prd). This means that if we want to build a model which predicts or explains the value of productivity, we need to consider fragment f2 as a potential building block. The actual specification of the relationship and its associated modeling assumptions represented by the arc <IT-Prd> can be looked up in the definition of fragment f2, which is shown below. In this case, we find the monotonic relationship Productivity = $M^+(IT)$, which holds if the four modeling assumptions

OntologyAssumption=influences, SimplifyingAssumption= qualitative, OperatingAssumption=quasi-static, and TimeScaleAssumption= medium are satisfied.

OKB CORPX

ALIASES

/Partnership, Pship/ /Product_Quality, PQual/ /Customer_Satisfaction, CSat/ /Customer_Service, CSrv/ /Marketing_Position, MPos/ /Promotional_Expenditure, PrmExp/ /Productivity, Prd/ /Information_Technology, IT/ /Revenue, Rev/ /Net Income, NInc/ /Production Cost, PCost/ /Performance, Perf/ /Goodwill, Gw/

END

ASSUMPTION CLASSES

/Ontology Assumption, OntAss/ (influences, cash flow, material flow); /Simplifying Assumption, SimpAss/ (qual, semi-qual, quant); /Operating Assumption, OpAss/ (static, quasi-static, dynamic); /Time Scale Assumption, TScAss/ (short, medium, long)

END

fragment f1 (IT) (Pship)

{qualitative model describing the relationship between IT and Partnership} relations

conditions

Partnership = $M^+(IT)$

OntAss=influences,

OpAss=quasi-static, TScale=medium

SimpAss=qual,

end

fragment f2 (IT) (Prd)

{qualitative model describing the relationship between IT and Productivity}

conditions

OntAss=influences, SimpAss=qual, OpAss=quasi-static, TScale=medium

relations

Productivity = $M^+(IT)$

end

fragment f3 (Prd) (Pship)

{qualitative	model	describing	the
relationship	between	Productivity	and
Partnership}			

conditions

OntAss=influences,	SimpAss=qual,
OpAss=quasi-static,	TScale=long

relations

Productivity = M^+ (Partnership)

end

.

fragment f18 (Price) (Sales)

relationship between Price and Sales volume}

conditions

OntAss=influences, SimpAss=qual, OpAss=quasi-static, TScale=medium

relations

Sales = $M^{-}(Price)$

end

fragment f19 (Price) (Sales)

{marketing model describing the semiquantitative relationship between Price and Sales volume}

conditions

OntAss=cash_flow, SimpAss=qual-OpAss=dynamic, TScale=short quant,

relations

Sales(t) = [68000, 92000] + [40000, 48000]*Price(t)

end

fragment f20 (Price) (Sales)

describing the {marketing model quantitative relationship between Price and Sales volume}

conditions

OntAss=cash flow, SimpAss=quant, OpAss=quasi-static, TScale=short

relations

Sales = 80000 - 44000*Price

end

fragment f21 (Sales) (Rev)

{marketing model describing the qualitative {accounting model describing the qualitative relationship between Sales volume and Revenue }

conditions

OntAss=influences, SimpAss=qual, OpAss=quasi-static, TScale=medium

relations

Revenue = M^+ (Sales)

end

end

fragment f22 (Price) (Rev)

{accounting model describing the qualitative relationship between Price and Revenue}

conditions

```
OntAss=influences, SimpAss=qual, OpAss=quasi-static, TScale=medium
```

relations

Revenue = M^+ (Price)

end

fragment f23 (Price, Sales) (Rev)

{accounting model describing the quantitative relationship between Price, Sales volume, and Revenue}

conditions

OntAss=cash_flow, SimpAss=quant, OpAss=quasi-static, TScale=medium

relations

Revenue = Price*Sales

end

fragment f24 (Rev) (NInc)

{accounting model describing the qualitative relationship between Revenue and Net Income}

conditions

OntAss=influences, SimpAss=qual, OpAss=quasi-static, TScale=medium

relations

NetIncome = M^+ (Revenue)

fragment f25 (Cost) (Price)

{financial model describing the qualitative relationship between Cost and Price}

conditions

OntAss=influences, SimpAss=qual, OpAss=quasi-static, TScale=medium

relations

 $Price = M^+(Cost)$

end

fragment f26 (PCost) (Cost)

the {financial model describing the qualitative ice, relationship between Production Cost and Total Cost}

conditions

OntAss=influences, SimpAss=qual, OpAss=quasi-static TScale=medium

relations

 $Cost = M^+(ProductionCost)$

end

fragment f27 (Cost) (NInc)

{financial model describing the qualitative relationship between Cost and Net Income}

conditions

OntAss=influences, SimpAss=qual, OpAss=quasi-static, TScale=medium

relations

NetIncome = $M^{-}(Cost)$

end				R-4:	
fragment f28	8 (Cost, Rev)	(NInc)			
{accounting	model	describing	the		
quantitative	relationship	between	Cost,	end	
Revenue, and	Net Income	}			

conditions

OntAss=cash_flow, SimpAss=quant, OpAss=quasi-static, TScale=medium

relations

NetIncome = Revenue - Cost

end

fragment f29 (Perf) (Gw)

{marketing model describing the qualitative relationship between Performance and Goodwill}

conditions

OntAss=influences, SimpAss=qual, OpAss=quasi-static, TScale=medium

relations

 $Goodwill = M^+(Performance)$

end

•

rules

R-1: OpAss(quasi-static)

R-2: SimpAss(qual) => solver(QSIM)

R-3: SimpAss(qual-quant) => solver(RCR)

Hence, if we are building a qualitative model describing, among other things, the impact (or influence) of IT usage on productivity, we must consider the inclusion of fragment f2 in the composite scenario model to be built.

In general, arcs emanating from a node x indicate the variables directly influenced by variable x. Thus, usage of IT has, in our enterprise model, a direct impact on Partnership, Productivity, and Customer Service. However, besides the direct influence of IT on Productivity, there is also an indirect influence of IT on Productivity, via Partnership. Indirect influences are represented in the interaction graph as a sequence of arcs called an *interaction path*. Here, the sequence <IT-Pship>-<Pship-Prd>, or more compactly written as <IT-Pship-Prd>, expresses the indirect influence of IT on Productivity. Similarly, IT has many more indirect influences on other variables, for example, the interaction paths <It-Prd-Perf-Gw> and <IT-CSrv-CSat-Gw> represent alternative possibilities of modeling the indirect influence of IT on Goodwill. Incoming arcs of a node x represent the direct influences on variable x. Our example indicates that Productivity is directly influenced by IT usage and Partnership. However, IT has a self-loop as the only incoming arc. The arc going from node IT back to itself means that the only influence on variable IT is IT itself, in other words, IT cannot be explained within the enterprise model. IT has to be determined outside of the model, that is, IT is treated as an exogenous variable whose value needs to be imported from a separate database when IT is included in a scenario model. Exogenous variables are typically variables which are, at least to some extend, controllable. The level of IT, for example, is determined by the budget proposed and passed by the management.

An arc label actually consists of a list of fragment identifiers. Such a list may be empty, as in the case of arc <IT-IT>, indicating an exogenous variable; may contain one identifier, as in <IT-Prd> meaning that the OKB knows only about one relationship between IT and Prd; or it may contain several identifiers suggesting alternative relationships. Two examples of multiple relationships are, first, arc <Price-Rev> which lists two fragments, f22 and f23, both using a relationship between price and revenue, and, second, arc <Price-Sales> which names three alternatives, fragments f18, f19 and f20, of modeling price and sales. Relationships involving more than two variables are identified by any of the participating variables. For example, fragment f28, which specifies a relationship between three variables net income (NInc), cost (Cost) and revenue (Rev), must be instantiated when either of the two arcs <Cost-NInc> and <Rev-NInc> is considered.



Figure 2: Interaction Graph of the CORPX Organizational Knowledge Base.

4. Model Composition

We refer to the real-world phenomenon under study as a scenario, and to the model representing it as a scenario model. Selecting the right model pieces to compose an appropriately integrated model for answering a given query requires modeling decisions along several dimensions. What is the best set of variables to be included in the model? What level of detail is appropriate? Which are the relevant organizational phenomena for studying the posed question? From what perspective should the problem be viewed? What kinds of approximations and abstractions should be allowed? Even the most carefully organized model fragment library won't provide enough information to find an answer to all of these questions. Therefore, we need to derive missing pieces of information from the query itself, that is, we need to look for clues provided in the query that could narrow the focus of the model composition process and reasonably constrain the set of plausible modeling assumptions.

As an example of composing a scenario model in response to a prediction question, let us suppose the user entered the query "How does an increase in price affect net income?" Conceptually based on natural language processing, a query elaboration

procedure would analyze the issued query and derive from it a set of ground expressions which would be passed on to the model manager module of the EMS for evaluation. In the absence of such a sophisticated query analyzer, we could simply devise a primitive query language which basically lists a number of ground expressions which permit the system to identify objects, quantities and relations of interest, where each of these has a referent in the organizational knowledge base. Hence, let us consider the simplified query {increase(Price), quantity(NetIncome)}, whose ground expressions increase(Price) and quantity(NetIncome) provide the input to the model manager.

The query indicates that we need a scenario model which computes net income. While the one ground expression quantity(NetIncome) of the query hints neither to a qualitative nor to a quantitative modeling approach, the other ground expression does provide a clear clue for a qualitative analysis. Since the increase operator indicates a desired direction of change without further specification, it suggests a qualitative model for investigating this effect on net income in the given scenario. Now, we could try to enumerate all possible combinations of model fragments, and to prune out those which either violate some of the modeling assumptions or prove to be irrelevant or insufficient regarding the query. However, this approach would be computationally too costly, considering the many combinations of model fragments, which would typically occur in an enterprise-wide environment. The number of modeling assumptions, on the other hand, tends to be much smaller, and therefore suggests a computationally better alternative by reasoning about combinations of modeling assumptions first, and then to select and integrate a suitable set of model fragments. We will come back to this computational issue below and discuss it in more detail.

Model composition begins with the derivation of an initial set of quantities of interest from the query. Quantities of interest correspond to variables which need to be included in the scenario model to be composed. In our example, we would take the query {increase(Price), quantity(NetIncome) and derive {Price, NetIncome} as the set of initial quantities of interest. The quantity operator in the second ground expression, {quantity(NetIncome)}, provides a hint that the value of the variable NetIncome is desired, which means that we are supposed to model a scenario wherein NetIncome is predicted. Variables to be predicted by a scenario model are called goal variables. The increase operator in the first ground expression, increase(Price), describes a manipulation to be performed on a variable. In this case we are supposed to (qualitatively) change the current value of price and then examine the effect of this change. Variables like Price which are to be changed initially in a way prescribed by manipulation operators in the query are called driving variables. Driving variables are used to drive the model building process by trying to establish a connection between them and the goal variables such that the values of the goal variables can be determined if an initial state

description is given. In the example, we would try to build a model that computes NetIncome from Price. In order to accomplish this job, the EMS employs a compositional modeling approach which searches the OKB for relevant model fragments which then can be used to construct an appropriate model.

Before invoking the model composition process, we need to define a modeling environment that selects a set of modeling commitments that are appropriate for the given query. Most importantly, this includes the selection of query-consistent modeling assumptions. In general, this is an indeterminate task. Ideally, we would choose the most suitable option along each of the modeling dimensions. However, we cannot expect that the query presented by the user conveys enough information to make indisputable decisions for all modeling assumption classes. For the sake of simplicity, let us suppose, in the example of this paper, that our query analyzer derives a uniquely determined modeling environment³. More specifically, we assume that

- (i) the encounter of the generic increase in the query implies a purely qualitative analysis,
- (ii) a qualitative analysis requires an ontological commitment to view the interactions in the enterprise as general influences between organizational variables,
- (iii) the organizational processes involving price and net income work at a medium time scale, and
- (iv) we restrict the analysis to quasi-static models (at least for now).

Hence, we would select

(OntAss=influences, SimpAss=qual, OpAss=quasi-static, TScale=medium)

as the current set of modeling assumptions.

Unquestionably, many queries would lead to ambiguous decisions about those modeling assumptions. On the other hand, it actually would be desirable in such cases that the EMS would generate a scenario model for all plausible modeling environments, that is, for all combinations of modeling assumptions that could

³ The user is supposed to have the option to define or redefine the modeling environment at any time, and thus can explicitly select particular modeling assumptions or override modeling assumptions chosen by the EMS.

reasonably be derived from the query. Furthermore, in order to promote user interaction and extend user control with the EMS, the query language should allow the user to explicitly select some of the modeling assumptions anyway. And finally, one of the modeling control parameters could provide a confirmation option that forces the EMS to merely suggest important modeling decisions, and to seek user confirmation before proceeding. This would be especially useful in ambiguous situations in which the user could prevent the system from generating, and subsequently solving, unnecessary or unwanted models.

In the first step, the driving and goal variables are located in the interaction graph depicted in figure 2. Our example has only one of each, the driving variable price and the goal variable net income (NInc). Now, it needs to be checked whether initial values of the driving variables are provided. Initial values could be derived from the query, supplied by an external data base, or computed from other variables. The latter is computationally the most expensive possibility because it entails constructing a more complex scenario model, and is thus eschewed unless the former two fail. Our example query has no clue on the initial value of price. Fortunately, the second possibility applies, because the node representing price has a self-loop. This means that the variable price can be treated as an exogenous variable, that is, its current value can be obtained from an external source. Next, we try to connect the driving variables with the goal variables, that is, we search the interaction graph for interaction paths between driving and goal variables. Looking at figure 2, there are four interaction paths describing different ways of computing net income from price. Each of the four generated interaction paths suggests to make use of a different collection of fragments for building a model that predicts how an increase of price would affect the net income of the CORPX enterprise.

	Interaction Path	# arcs	# nodes	# frags	# models
1	Price-CSat-Gw-MPos-Sales-Rev-NInc	6	7	8	4
2	Price-CSat-MPos-Sales-Rev-NInc	5	6	7	4
3	Price-Sales-Rev-NInc	3	4	7	12
4	Price-Rev-NInc	2	3	4	4
	Total				24

 Table 1: Combinations of Different Interaction Paths.

Potential scenario models, or *model candidates*, differ in their complexity measured in terms of number of variables and number of fragments involved in composing them. From table 1 we can see that the first interaction path relates

seven variables by six arcs identifying eight relationships represented in eight fragments. Since some of the arcs suggest a set of alternative relationships, we can choose from several different combinations of relationships and their associated fragments. As another example, the third interaction path in table 1, Price-Rev-NInc, consists of three arcs <Price-Sales>, <Sales-Rev>, and <Rev-NInc> which suggest the sets { f_{18} , f_{19} , f_{20} }, { f_{21} , f_{23} }, and { f_{24} , f_{28} } as possible fragments for modeling relationships between respectively price and sales, sales and revenue, and revenue and net income. Thus, any fragment triple

$$\langle F_1, F_2, F_3 \rangle \in \{ f_{18}, f_{19}, f_{20} \} \times \{ f_{21}, f_{23} \} \times \{ f_{24}, f_{28} \}$$

is an eligible candidate for a scenario model, resulting in 3*2*2 = 12 combinations to choose from. Likewise, we can produce yet more candidate models, and represent them as fragment n-tuples where n denotes the number of participating fragments, from the other interaction paths. Specifically, the first interaction path generates four 6-tuples, the second four 5-tuples, and the last four pairs of fragments. All together, table 2 lists a total of 24 candidates to consider when building a model of the scenario described in the given query "*How does an increase in price affect net income?*" Obviously, the number of model candidates varies with every query, and, in more intricate scenarios, can quickly reach an order of magnitude that is hard to manage.

In order to keep the model composition task tractable we would like to avoid a complete enumeration of possible model candidates. Since the final scenario model has to be internally consistent, we must eliminate those candidates from further consideration whose constituting fragment's precondition sections contain contradictory modeling assumptions because this would indicate an incompatible set of model fragments. In our example above, we have hitherto ignored the modeling assumptions which the fragments are based upon. Prior to generating candidate models, we need to check the consistency of modeling assumptions. From the current modeling environment, we obtain the active set of modeling assumptions, (OntAss=influences, SimpAss=qual, OpAss=quasi-static, TScale=medium) whereon the building of the scenario model rests.

	IP	Model Candidate	Incompatible Fragments
1	1	$(f_{12}, f_8, f_9, f_{14}, f_{21}, f_{24})$	0
2	1	$(f_{12}, f_8, f_9, f_{14}, f_{21}, f_{28})$	(f ₂₈)

3	1	$(f_{12}, f_8, f_9, f_{14}, f_{23}, f_{24})$	(f ₂₃)
4	1	(f ₁₂ , f ₈ , f ₉ , f ₁₄ , f ₂₃ , f ₂₈)	(f ₂₃ , f ₂₈)
5	2	$(f_{12}, f_{10}, f_{14}, f_{21}, f_{24})$	0
6	2	$(f_{12}, f_{10}, f_{14}, f_{21}, f_{28})$	(f ₂₈)
7	2	$(f_{12}, f_{10}, f_{14}, f_{23}, f_{24})$	(f ₂₃)
8	2	$(f_{12}, f_{10}, f_{14}, f_{23}, f_{28})$	(f ₂₃ , f ₂₈)
9	3	(f ₁₈ , f ₂₁ , f ₂₄)	0
10	3	(f ₁₈ , f ₂₁ , f ₂₈)	(f ₂₈)
11	3	(f ₁₈ , f ₂₃ , f ₂₄)	(f ₂₃)
12	3	(f_{18}, f_{23}, f_{28})	(f ₂₃ , f ₂₈)
13	3	(f_{19}, f_{21}, f_{24})	(f ₁₉)
14	3	(f_{19}, f_{21}, f_{28})	(f ₁₉ , f ₂₈)
15	3	(f_{19}, f_{23}, f_{24})	(f ₁₉ , f ₂₃)
16	3	(f ₁₉ , f ₂₃ , f ₂₈)	(f_{19}, f_{23}, f_{28})
17	3	(f_{20}, f_{21}, f_{24})	(f ₂₀)
18	3	(f_{20}, f_{21}, f_{28})	(f ₂₀ , f ₂₈)
19	3	(f_{20}, f_{23}, f_{24})	(f ₂₀ , f ₂₃)
20	3	(f_{20}, f_{23}, f_{28})	(f_{20}, f_{23}, f_{28})
21	4	(f ₂₂ , f ₂₄)	0
22	4	(f ₂₂ , f ₂₈)	(f ₂₈)
23	4	(f ₂₃ , f ₂₄)	(f ₂₃)
24	4	(f ₂₃ , f ₂₈)	(f ₂₃ , f ₂₈)

Table 2:Model Candidates generated from Interaction Paths (IP) using
(OntAss=influences, SimpAss=qual, OpAss=quasi-static, Tscale=medium)
as the currently active set of modeling assumptions.

Model Candidate	Model Size
(f ₁₂ , f ₈ , f ₉ , f ₁₄ , f ₂₁ , f ₂₄)	13
(f ₁₂ , f ₁₀ , f ₁₄ , f ₂₁ , f ₂₄)	11
(f_{18}, f_{21}, f_{24})	7
(f ₂₂ , f ₂₄)	5

Table 3: Remaining models after eliminating incompatible models.

Model size measured in terms of the candidate evaluation function: eval(m)=v+r.

Table 2 shows for each model candidate those fragments that are incompatible because they violate some of the active modeling assumptions. Notice that only four (set in boldface) out of twenty-four model candidates are indeed internally consistent. Therefore, we devise a compositional modeling strategy which reasons first about the consistency of the modeling assumptions before it starts to assemble

composite model candidates. This approach reduces quickly the search space of possible scenario models from 24 to just 4 candidates (see table 3), namely (f_{12} , f_8 , f_9 , f_{14} , f_{21} , f_{24}), (f_{12} , f_{10} , f_{14} , f_{21} , f_{24}), (f_{18} , f_{21} , f_{24}), and (f_{22} , f_{24}). Thus, our compositional modeling method concludes, for this example

{increase(Price), quantity(NetIncome)} ==> or ((f_{12} , f_8 , f_9 , f_{14} , f_{21} , f_{24}), (f_{22} , f_{24}) (f_{12} , f_{10} , f_{14} , f_{21} , f_{24}), (f_{18} , f_{21} , f_{24})).

5. Candidate Evaluation and Model Integration

After generating a set of feasible scenario model candidates we need to evaluate and order them according to their appropriateness in order to choose the most appropriate one as the final scenario model. Unfortunately, there is no single criteria that would alone by itself describe appropriateness in a satisfying way, which makes it difficult to provide a definition of it that is not arbitrary to some extend and, at the same time, operational. Falkenhainer and Forbus (1991) define the final scenario model as the model candidate which is coherent and most useful. The former criterion requires the scenario model to be consistent with the modeling assumptions to which the EMS committed when exploring the scenario set up by a query. The latter refers to the tradeoff between information cost and significance to the query in terms of sufficiency and minimality (Balakrishnan and Whinston 1991). Sufficiency means that the answer to the query is firstly not only correct but also relevant to the question such that it provides her with the information sought, and secondly that the answer is also satisfactorily detailed and accurate. Minimality, on the other hand, calls for a parsimonious response and forbids elaborate details. In other words, we are looking for the scenario model which is minimal and (a) consistent, (b) valid, (c) relevant, (d) adequately detailed, and (e) adequately accurate.

The candidate evaluation module receives, from the model composition module as its input, a set of scenario model candidates, and produces as its output the final scenario model. First, we need to ensure that the scenario model satisfies restrictions (a) to (e). Fortunately, our compositional modeling method was designed such that it generates only feasible model candidates, that is, candidates which do satisfy the above restrictions. Consistency is accomplished through the explicit reasoning about underlying modeling assumptions during the model building process. We can assume validity of the relationships used in building model candidates because model fragments are only applied if the assumptions stated in their preconditions section hold. We ensure relevance by assuming that our query analyzer identifies quantities of interest correctly, and that only such fragments are considered which the interaction graph relates to a driving variable or a goal variable. An adequate level of detail and accuracy is achieved by assuming that the query analyzer, in connection with user interaction, is able to
derive and establish a proper set of modeling assumptions which include an appropriate description of level of detail and accuracy required in the given scenario.

Now that we are assured that all remaining candidate models are indeed feasible and appropriate in the sense that we can expect them to yield comparably satisfactory answers, we want to select the minimal or simplest one. Let us define simplicity of a model in terms of model size and define an evaluation function eval(m) as a function of number of variables v and number of relationships r, for example as eval(m) = v + r. From table 3 we can see that candidate (f₂₂, f₂₄) is chosen as the final scenario model in our example and is passed on to the solver for model execution.

Using candidate model (f_{22}, f_{24}) we obtain

scenario model f22-f24 (Price) (NInc) conditions OntAss=influences, SimpAss=qual, OpAss=quasi-static, TScale=medium relations Revenue = M⁺(Price) NetIncome = M⁺(Revenue) end

as the final model, the scenario model. This is not yet a completely specified model, one which could be solved as it is. However, it does show the complete specification of the model's constraints, the core part of the final scenario model. Although it contains only valid relationships, a price increase leads to higher revenues and highter revenues have a positive effect on net income, it may not be accurate enough for the user's current analysis. Model (f_{22} , f_{24}) ignores, for example, the influence of price changes on sales which in turn also influence the goal variable net income. Hence, after inspecting the suggested scenario model, the user may opt for rejecting it, and thus force the EMS to search for an alternative scenario model. In our example, the EMS would suggest model candidate (f_{18} , f_{21} , f_{24}) as its next scenario model.

Revenue = M⁺(Sales) NetIncome = M⁺(Revenue)

end

The new, more complex model does represent the indirect effect of price on net income by including sales as an additional variable and by adding another relationship. Depending on the user's intents, it may be beneficial to trade off some model cost (in terms of model complexity) for more accuracy.

7. Conclusion

To conclude, we have presented a novel, comprehensive framework for future enterprise-wide modeling systems in this paper. Enterprise computing is a support tool to achieve organizational goals. We believe that future research in model building and model management for decision support in organizational environments requires more attention to organizational knowledge representation and to related model building and model reasoning research in artificial intelligence. One purpose of this paper is to bring to bear some of the stimulating results obtained from the AI community, and to indicate how they can be incorporated into the DSS research on model building. Among the new features we have proposed, we want to highlight those which, in our mind, map out the most promising future research directions. First, the possibility of both qualitative and quantitative model formulations, which introduces a new level of versatility to organizational model building, and which should widen the scope of computer supported decision tools considerably. Second, the explicit representation of modeling assumptions. And finally, the application of a compositional modeling strategy to automatically build task-specific scenario models, which liberates users from having to specify special modules for controlling the modeling integration process.

References

- Addanki, S., R. Cremonini, and J.S. Penberthy (1991) "Graphs of Models." <u>Artificial Intelligence.</u> **51**: 145-178.
- Balakrishnan, A., and A.B. Whinston (1991) "Information Issues in Model Specification." Information Systems Research. 2(4): 263-286.
- Basu, A. and R. Blanning (1994) "Model Integration Using Metagraphs." <u>Information Systems Research</u>. **5**(3): 195-218.
- Bhargava, H.K. and R. Krishnan (1991) "Reasoning with Assumptions, Defeasibility, in Model Formulation." working paper, The H. John Heinz III School of Public Policy and Management, Carnegie Mellon University.

- Bonczek, R.H., C.W. Holsapple, and A.B. Whinston (1981) Foundations of Decision Support Systems, Academic Press.
- Carter, G.M., M.P. Murray, R.G. Walker, and W.E. Walker (1992), <u>Building</u> <u>Organizational Decision Support Systems</u>, Academic Press, San Diego, CA.
- De Kleer, J., and J.S. Brown (1984) "A Qualitative Physics Based on Confluences." <u>Artificial Intelligence</u>. 24: 7-83.
- Dolk, D. R., and J.E. Kottemann (1993) "Model Integration and a Theory of Models." <u>Decision Support Systems</u>. 9(1): 51-63.
- Falkenhainer, B., and K.D. Forbus (1991) "Compositional Modeling: Finding the Right Model for the Job." <u>Artificial Intelligence</u>. **51**: 95-144.
- Geoffrion, A.M. (1987) "An Introduction to Structured Modeling." <u>Management</u> <u>Science</u>. **33**(5): 547-588.
- Hinkkanen, A., K.R. Lang, and A.B. Whinston (1995) "On the Usage of Qualitative Reasoning as Approach Towards Enterprise Modeling." <u>Annals</u> <u>of Operations Research.</u>, in press.
- Krishnan, R., P. Piela, A. Westernberg (1991) "On Supporting Reuse in Modeling Environments." working paper, School of Urban and Public Affairs and Engineering Design, Carnegie Mellon University.
- Kuipers, B. (1988) "Qualitative simulation using time-scale abstraction." <u>Artificial</u> <u>Intelligence in Engineering</u>. **3**(4): 185-191.
- Liang, T. (1988) "Development of a Knowledge-Based Model Management System." <u>Operations Research</u>. **36**(6): 849-863.
- Mannino, M., B. S. Greenberg, and S. N. Hong (1990) "Model Libraires: Knowledge Representation and Reasoning." <u>ORSA Journal on Computing</u>. 2(3): 288-301.
- Muhanna, W.A., and R.A. Pick (1992) "Meta-Modeling Concepts and Tools for Model Management: A Systems Approach." forthcoming in <u>Management</u> <u>Science</u>.
- Petrie, C. (Ed) (1992) Enterprise Integration Modeling: Proceedings of the First International Conference. MIT Press.
- Raghunathan, S., R. Krishnan, and J. H. May (1993) "MODFORM: a Knowledge-Based Tool to Support the Modeling Process." <u>Information Systems</u> <u>Research.</u> 4(4): 331-358.
- Rickel, J. and B. Porter (1992) "Automated Modeling for Answering Prediction Questions: Exploiting Interaction Paths." In Proceedings of the Sixth

International Wrokshop on Qualitative Reasoning. p. 82-95. Edingburgh, Scotland.

Weld, D.S. (1992) "Reasoning About Model Accuracy." <u>Artificial Intelligence</u>. **56**: 255-300.

A Field Service Support System Using the Computer Analysis of Networks of Queues (CAN-Q) Model

H.T. Papadopoulos and J. Darzentas

Dept. of Mathematics, University of the Aegean, GR-832 00 - Karlovasi, Samos, Greece

The Field Service (FS) Organization of many companies constitutes a vital Department playing an important role to their success. Field Service managers need some tools to analyse the impact of their decisions on customer service level, which must be as high as possible (at least 95%), and inventory cost, trying to reduce it to a minimum level (its ratio to the total FS revenue must be less than 10%). Such a tool is presented in this paper. We applied a closed queueing network model developed by Solberg, called CAN-Q (acronym for Computer Analysis of Networks of Queues). This model, although originally developed for modelling Flexible Manufacturing Systems, has been applied to the FS organization of a subsidiary of multinational computer company, in Greece, and it has proved to be very efficient from the computational point of view and constitutes a powerful tool for the FS managers, providing them with some useful performance measures. Its successful application gives rise to try some other queueing network models, available from the literature, in combination with various inventory control models to help FS managers solve their critical problems.

Keywords: *Strategic planning; queueing networks; field service; decision support systems.*

1. Introduction

Many industries, multinational and national companies set as their first priority the so-called 'customer satisfaction', as they know that only in this way they can maintain or even increase their market share. There are many factors that contribute to this end. To name a few, the quality of the products they sell, the quality of the service they provide, both administrative and technical, among others. The FS Department has the responsibility of the service maintenance of all the different products, which are sold by sales forces of the company and which many times

may come from different manufacturers, making the management of the field service a difficult task. Field Service managers have to cope with the two conflicting objectives: (a) to maintain high level of customer service, which must be above 95%, and (b) keep the spares inventory level as low as possible. An efficient measurement for this, is to keep the ratio of the spares inventory cost to the total revenue of the FS organization less than 10%, usually between 5%-8%. Some other important decisions that the FS manager has to take are how many FS engineers to have and how to allocate them to the company's customer population.

In this paper, we attempted to approach the FS manager's problem by implementing the computer analysis of networks queues (CAN-Q) model, which is a closed queueing network (CQN) model introduced by Solberg [5] mainly to model the flexible manufacturing systems (FMS) (for a detailed analysis of FMS as CQNs the interested reader is addressed to Buzacott and Shanthikumar [1], Chapter 8, Gershwin [3], Chapters 6 and 9, and to Papadopoulos *et al.* [4], Chapter 3, among others). The idea of modelling the field service problem as a closed queueing network is justified as the number of customers of a FS organization is usually constant, like the number of parts/pallets circulating within a FMS. Further, we decided to implement CAN-Q and not any other CQN model, in order to exploit the existing software programme, developed by Solberg's research team. We applied this model to the data of the FS organization of a multinational computer subsidiary, in Athens, and we saw that it performs quite satisfactorily. Unfortunately, due to confidentiality reasons, numerical data and results cannot be presented.

This paper is organized as follows. In the next section, we present the structure of a real FS organization having in mind that of the (multinational) computer subsidiary, and we make some remarks relative to the function and operations of this organization. Then, in the following section, we give the development of the model. Last section concludes the paper and recommendation for further research is suggested. Finally, in the Appendix, the analysis of the CQN model is given with the implementation of the CAN-Q algorithm (Solberg's model).

2. The Structure of a FS Organization of a Computer Company

In this section, we describe shortly the structure and the operation of a FS organization of a computer (Multinational) subsidiary, as this motivated and formed the basis for the development of our FS support system. We believe that the structure and operation of the FS Departments of other industries are quite similar (e.g. photocopier, communication companies, etc.). For this reason, we strongly recommend the application of the proposed model, described in the next section, to these companies as well, with slight modifications and adjustments. In a computer (subsidiary of a multinational, or national of medium size) company, the FS organization consists mainly of two Departments:

- (a) the FS contracts Sales Department, which brings revenue to the organization by selling service contracts and other services (such as preparation of computing rooms to accommodate the computing systems), to the customers. Other source of revenue is the revenue transfer coming from the Sales Department as a certain percentage on the sales of hardware (H/W) and Software (S/W) products, covering the one-year period during which the products are under guarantee, and
- (b) the Technical Support Department, which consists of the engineers (technicians), for both H/W and S/W products. This Department together with the administrator, the secretaries, the call hander(s), the spare parts inventory, the various travelling and training expenses, and of course the wages of all the FS employees, constitute the expense part of the FS organization.

The operation of the technical support Department is done as follows. The customer who has a problem calls the call-handler in the FS Support Department and this person longs the call in the so-called FS Log-Book, checks the contract of the customer and passes the information to the dispatcher who arranges for an engineer to take care of this call, according to the terms and conditions of the contract of this customer. There are various classes of customers, depending on the type of their contract with the company. A typical classification, for example, is:

- 1. *Class-1 customers:* These are customers with 24-hour coverage and they are assigned the highest priority, the company being obliged to service these customers immediately, having an engineer stand-by, the non-working hours, even on the weekends and holidays. Example of such a customer is a production plant (e.g., cement industries, etc.) which maintains 3 shifts per 24 hours.
- 2. *Class-2 customers:* These are high-priority customers too, where according to the contract, the FS engineer must go and fix their problem within 2 hours from the call. A representative example of such a class of customers is a bank which wants to maintain a reliable on-line system.
- 3. *Class-3 customers:* These are also priority customers, where according to their contract, the FS engineer must go and fix their problem within 4 hours from the call. An insurance company may belong to this class of customers.
- 4. *Class-4 customers:* These are the normal customers, where according to their contract, the FS engineer must go and fix their problem within 8 hours from the call. The majority of the customers population of the company belong to this class (e.g. Universities, research institutes, various private commercial companies, etc.).

66 A Field Service Support System Using the Computer Analysis of Networks of Queues

- 5. *Class-5 customers:* These are customers with an *elementary* service maintainance contract, that cannot afford a normal contract and try to be covered somehow at least from the spare parts point of view, where according to their contract, the FS engineer must go and fix their problem within 16 hours from the call. Of course, the cost for such a contract is lower than a normal (say 65% of the price of a normal contract). The company tries not to sell such contracts and for this, only a few customers belong to this class.
- 6. Class-6 customers: These are customers without contract, the so-called *per-call customers*. The rules of the company dictate the engineers to give them very low priority and depending on the work-load, they are obliged to go and fix their problem within a week, over-charging them, of course, in order to force them sign a normal service maintainance contract.

The FS manager together with the FS Unit managers (managers of the H/W and S/W engineers) are obliged to service all these customers trying to always be consistent with the terms and conditions of their contract.

One of the problems they face is how to allocate the engineers to the various customers. Usually the allocation is done depending on the account (customer class) and the type of the system (super, mini, PCs-UNIX, DOS, VMS, etc.) and the part of the computer (CPU, peripherals (discs & tapes), printers, terminals, etc.). It is very common the case where more than one engineers are specialized on a specific area or part of the computer or type of operating system, for back-up reasons and for easily dispatching the calls among them.

FS managers, restricted by the budget, don't have the luxury of hiring new engineers. Instead, they invest more money on training the existing staff in different areas (e.g., both for mini systems and PCs, and H/W and S/W products) in order to decrease the idle time of the engineers and increase their utilization, efficiency and productivity, in general.

Concerning the inventory investment and the spare parts stock management, this is an important factor affecting the probability an engineer will not have the necessary parts. In this work, it's not our aim to deal with stock control models. We just make some remarks.

To handle all the various classes of customers, from the stock control point of view, we say that, first of all, this is the Logistic manager's (the materials manager's, more specifically) responsibility. The manufacturer, which in a multinational company, is the 'mother'-company, issues special Logistic and Field Service plans for all the products, providing a *recommended spares list* (RSL) for various levels of service (LOS), usually 95% and 98%. Then, depending on the consumption of the spares (forecasted consumption index for the new products,

based on the MTBF given by the Quality Department of the manufacturing plant), the re-ordering policy is chosen, by setting the appropriate target stock levels (TSL) for each line item. In either case, the priority of the customers is always taken into account. For example, for class-1 customers, it's not rare to keep dedicated stock, which depending on the distance of the customer's site, it may be stocked at customer's place, after negotiations with the customer. For class-1 and class-2 customers, the so-called *option-swap material* solution is also applied. For example, if the customer is a production plant or a bank and the problem cannot be fixed within 1-2 hours, the FS engineer replaces the whole CPU unit or the peripheral, by a similar one, stocked for this purpose and for this particular customer, at the warehouse (W/H), and after the repair - which takes place at the company's premise, - the engineer returns the customer's option and takes back the company's one; this is the option-swap material. In this way, the customer doesn't remain 'down' for a long time and he/she is satisfied with this solution.

A critical point, tested experimentally, for a couple of years, at the Greek subsidiary of the computer company, is to *demolish the kits* and maintain single line items stock, in order to increase the availability of the spare parts and thus the level of service to the engineers (and to the customers). Of course, some kits must be maintained, for certain products that are highly consumed and their spares are highly demanded, e.g., for specific models of printers, terminals, etc.

Another important issue related to the Logistics and Field Service strategy, is to establish a *local repair center* at Company's premises (close to the warehouse), or at the sub-contractor's site. In this way, the lead time of a spare part's order is reduced greatly, reducing the spares stock level, to some good extent. Of course, some cost is involved here and this constitutes another decision variable for the FS manager. It is suggested the establishment of such a local repair centre, at some minimum cost, in the beginning, by not hiring new engineers, but utilizing the existing ones, on a rotational basis, starting with the repair of peripherals and PCs. Further investigation is needed on whether is beneficial to further invest on the repair of more complicated options (e.g., CPUs), for which expensive tools and equipment are needed. Data from other more developed subsidiaries of the same computer company (e.g., in Israel and Italy) supports this investment combined with the subcontracting of the engineering repair staff, due to the restrictions in hiring new employees, directed by the Headquarters.

To increase the utilization of the FS engineers, decreasing their idle time, is something that can be done at some cost in two ways: (i) keeping extra spares *stock on vans* that are moving in various areas of the city (where the most critical customers are), - this is the *mobile stock*, - paying money for security, as well, and (ii) having *dedicated courier service* at the warehouse, delivering in this way the spares at the customer allowing the engineers to move from customer to customer, without being obliged to visit the W/H to collect the good spares, for the new call,

or to return the defectives consumed at a former call. This solution is very useful, since otherwise, in cities like Ahtens, with heavy traffic, the time of the engineers is splitted almost equally at travelling and repairing!

3. Model Development and Application

The model we decided that it best suits to the FS organization of the computer company, described in the previous section, is the CAN-Q, a closed queueing network (CQN) model introduced by Solberg [6]. Of course, this model was developed primarily to model FMS with one type of 'customers'. We modified it appropriately to reflect better the needs of our FS organization. This model is illustrated in Figure 1.



Figure 1: A CQN model for the FS organization

The actual layout of this network has as follows. The customer population of the company is splitted into R different classes of customers (in our case R=6), depending on either the terms and conditions of the service maintainance contract or the type of the system (CPU, memory boards, discs/tapes, terminals, PCs, etc.).

It is the FS unit manager's responsibility to allocate the FS engineers among the customers, depending on their experience, training, knowledge, etc.

In Figure 1, stations 1,2,...,R, represent the customers of class 1,2,...,R, respectively, that have failed. Note R+2 models all the 'up'-customers of all the classes. Whenever an 'up'-customer goes down, he/she calls the company, more specifically the call-handler who logs the call at the so-called FS Log Book and informs the FS dispatcher (the FS Unit manager usually plays this role), who in turn allocates an available and appropriate engineer (depending on the customer class and the type of the problem). The engineer then goes to the customer's site to fix the problem, having first either visited the warehouse (W/H) or called the W/H attendant to bring him/her the spare part(s) he/she needs for the job, via a courier (this is represented by station R+1 in figure 1). For modelling purposes, when an 'up'-customer goes down, he/she informs the call-handler and he/she is transferred to one of the R stations - to receive service by a FS engineer, with probability q_i, (i=1,...,R),- depending on his/her class (q_{R+1} represents the probability that the customer becomes 'up'). After having received the (repair) service the 'down'customer becomes an 'up'-customer again and he/she is transferred to the R+2 station via the transporter station (node R+1). In reality, customers are at their sites and the courier transfers the good spare parts at their sites and after the service completion, the courier brings the defective parts back to the warehouse, plus any good parts that were not used for that particular call.

It has to be noticed that with the CAN-Q model, the 'up'-customers (node R+2) are not modelled. But it is very easy to do that and derive, say, the expected number of 'up'-customers, by the formula:

$$E[N_{R+2}] = N - \sum_{i=1}^{R+1} E[N_i],$$

where N denotes the total number of customers, which is known to the company and $E[N_i]$'s are calculated from the model (see the Appendix for the analysis of the CAN-Q model).

Stations 1,...,R+1, may be modelled as single-server or multi-server nodes, depending on the work-load of a particular class of customers (e.g., class-5 and class-6 customers are only a few and the respective stations are not only single-server stations but even they are pooled to justify the FS engineer utilization, and this facility (sensitivity analysis) is provided by the CAN-Q model).

The reason we applied the CAN-Q algorithm as a FS support system, was not only to exploit the S/W programme, which is available in FORTRAN, but also because the running time for solving typical realistic problems is very low (a few seconds only) and neither storage requirements nor execution time pose any difficulties. The output of this model includes the relative utilization and the station (i.e. FS

engineer) utilization, the expected number of customers of any class (at each node), the average number of customers (of any class) waiting in queue for service, the average waiting time spent in the system (for repair) by any 'down'-customer, etc.

Remarks: Nodes 1 and 2 that model the repair operation of the high priority customers may be modelled as self-service queueing systems, i.e., $M/M/\infty$. The only case where these two classes of the highest priority customers are not served immediately, is when the spare part (which is always available at the W/H or at customer's site) is dead-on-arrival (DOA), i.e., although is brand-new and sealed, there is a problem from manufacturing. This case is very rare, and the FS manager tries to find a solution for the critical customer to resolve the problem, implementing any unusual method he/she can imagine!

All the rest classes of customers (3-6) are served at nodes 3-6, respectively, which are modelled as M/M/c/K queueing systems, with the customers being serviced according to the FCFS service discipline.

Node R+2 is a fictitious queueing station accommodating all the 'up' customers of the company (something like the 'negative' customers), consisting of customers of all classes whose system/option is up and running. Notation concerned with the mean service rates (or mean service times) of all the nodes of the CAN-Q network (of Figure 1), as well as the definition of the various performance measures and the solution of the model, are given at the Appendix.

Applying the data of the Greek subsidiary of the multinational computer company, using the proposed CQN model, we were able to derive numerical results with a remarkable accuracy, - compared against the real data (the deviation varied from 2% to 13%), - concerning the various performance measures, such as the expected number of customers at each node and the expected utilization of the FS engineers, or equivalently the expected idle time of the FS engineers, and the production rate or equivalently the average waiting time spent by any customer in the system (meaning the total time elapsed from the time the customer called the call-handler till the time his/her problem was fixed). We discovered, based always on the real data, that for class-6 (per-call) and class-5 (with an elementary service contract) customers, the results for the FS engineers utilization were very very low, meaning that their idleness was very high, and this means that the FS manager must not allocate engineers only for these classes of customers. And this happens, indeed, in reality.

The advantage of the proposed model is that it models separately the transporter (courier), which is very realistic, as travelling time especially in cities with heavy traffic (like Athens) is not negligible at all.

4. Conclutions and Further Research

Applying the computer analysis of networks of queue (CAN-Q) algorithm, introduced by Solberg [6], we developed a closed queueing network model to form the basis for field service support system, by slightly modifying the CAN-Q model which was originally developed for modelling FMS. Running this model with the data of the FS organization of a subsidiary of a computer multinational company in Athens, Greece, we saw that it gives very accurate results. The proposed model offers the FS manager the possibility to estimate various useful performance measures such as the mean sojourn (repair) time of a customer of any class, the average production rate (how many customer calls are handled and completed per certain unit time), the utilization of the FS engineers, or equivalently their idleness, the expected number of customers (of any class) that are 'down' at any time and the maximum number of customers that may be allocated per FS engineer to maintain a high level of customer service (usually above 95%).

Although we focused on the application of the proposed CQN model to the small Greek subsidiary of the computer (multinational) company, we strongly believe that this model is applicable to many other companies with similar structure of their field service organizations, such as the photocopier and the (tele)communication industries, among others. The successful application of CAN-Q as a FS support system, give rise to try some other queueing network models, available from the literature, in combination with various inventory control models to help FS managers solve their critical problems.

A quite interesting and useful area for further research would be the development of a *simple, easy to use* (by the FS manager who usually doesn't know anything about queueing theory), *total system cost model*, incorporating all the possible decision variables into it, such as the number of the FS engineers required, their allocation to the various customer classes, the effect of the spares stock policy on the level of service (LOS), the FS engineers utilization or the percentage of their idle time, the cost for training them, and how much time a customer waits from the time of a breakdown till his/her problem has been fixed (total sojourn/repair time), and if the terms and conditions of his/her contract are being met, etc.

Appendix

Here, we first give the notation used in the CAN-Q network and the (mathematical) definition of the various performance measures that are derived from this model, and then we just present the solution of this model adopted to the FS organization of the computer company. The details of their derivation are omitted as these may be found in Solberg [5] & [6].

Notation:

N = the total number of customers (of all R = 6 classes) in the system (both 'up' and 'down');

 N_{R+2} = the number of 'up'-customers (of any class i=1,...,R);

 N_i = the number of 'down'-customers (of any class i=1,...,R) that are waiting for or in repair at the respective i-station;

 c_i = the number of servers (FS engineers) at station i (i=1,...,R,R+1);

 \hat{i}_i = the average service (repair) rate of station i (i=1,...,R,R+1), when it is busy (i.e., $1/\hat{i}_i$ is the average processing (repair) time at station i);

 $q_i =$ [For modelling purposes] the probability that the courier transfers the customer of class i (i=1,...,R) to station i for repair. [In reality] q_i is the probability the courier delivers the spare part(s) to the customer of class i (at station i). This probability, q_i , is nothing else but the failure frequency of a customer of class i;

 q_{R+1} = the probability that the 'down'-customer of any class i (i=1,...,R) has been repaired, becoming an 'up'-customer;

Definition of some performance measures:

• The *utilization of the ith station*, denoted by u_i, is defined as, the long run average number of busy FS engineers at station i. Mathematically, this is given by

$$u_i = \frac{q_i \cdot i_{R+1}}{i_i} u_{R+1}$$
, for all $i \neq R+1$ (i=1,...,R).

The *utilization per FS engineer* of the ith station is defined as the fraction of time that each FS engineer is busy, and is given by u_i/c_i .

• The *relative utilizations* of the i stations, denoted by r_i, are defined by the fractions appearing in the above formula, i.e.,

$$r_{i} = \frac{q_{i} \cdot \hat{1}_{R+1}}{\hat{1}_{i}} = \frac{u_{i}}{u_{R+1}}, \text{ for } i=1,...,R,$$

= 1, for i=R+1.

• The *production rate* of the FS Department (concerned with the repair branch) is defined as the steady-state average number of 'down'-customers that are being repaired per time unit (e.g., per day or week or month). This is equal to the mean rate of flow of repaired customers out of the transporter station (courier, i.e., station R+1). This is denoted by X and is expressed by the formula

$$\mathbf{X} = (\mathbf{q}_{\mathbf{R}+1} \cdot \mathbf{\hat{i}}_{\mathbf{R}+1}) \cdot \mathbf{u}_{\mathbf{R}+1}.$$

• The *average time spent in the system* by any 'down'-customer, denoted by W, is defined as the average total repair time, including transportation and any queueing time (waiting for the FS engineer to come), and is given by the well-known Little formula:

$$W = \frac{N}{X}$$
.

The solution:

The state of the system modelled by the network depicted in Figure 1, is indicated by the vector $(N_1,...,N_{R+1})$, and the steady-state probabilities, for single-server stations, are given by

$$P(N_1,...,N_{R+1}) = \frac{1}{G(R+1,N)} r_1^{N_1} \cdot r_2^{N_2} \cdots r_{R+1}^{N_{R+1}}.$$

where G(R+1,N) is a normalized constant defined so that all of the probabilities $P(N_1,...,N_{R+1})$ sum to one. Buzen [2] provided a very efficient recursive technique for its calculation.

If one or more of the stations has multiple FS engineers, a slight modification is necessary. For any such station i, having c_i FS engineers, the term in the product corresponding to that station (that is, $r_i^{N_i}$) must be replaced by $r_i^{N_i} / A_i(N_i)$, where,

$$A_{i}(N_{i}) = \begin{cases} N_{i}! & \text{if } N_{i} \leq c_{i}, \\ \\ c_{i}!c_{i}^{N_{i}-c_{i}} & \text{if } N_{i} > c_{i}. \end{cases}$$

The utilization of the transport station (courier), u_{R+1} , is of particular interest. This is given by (for a proof see Solberg [6])

$$u_{R+1} = \frac{G(R+1,N-1)}{G(R+1,N)}$$
.

As soon as u_{R+1} is determined, the average production rate of the FS Department, concerned with the repair operations, the expected time spent in the system by any 'down'-customer (of any class) and the utilizations of the stations are given, respectively, by

$$\begin{split} X &= q_{R+1} \cdot \hat{\imath}_{R+1} \cdot \frac{G(R+1,N-1)}{G(R+1,N)} , \\ W &= \frac{N \cdot G(R+1,N)}{q_{R+1} \cdot \hat{\imath}_{R+1} \cdot G(R+1,N-1)} , \end{split}$$

74 A Field Service Support System Using the Computer Analysis of Networks of Queues

$$u_i = r_i \frac{G(R+1,N-1)}{G(R+1,N)}$$

The marginal probability distribution for the number of customers at the ith station (i-class customers), provided that it is a single-server station, is given by

$$P(N_i = v) = \frac{r_i^{v}}{G(R+1, N)} [G(R+1, N-v) - r_i G(R+1, N-v-1)].$$

Station R+1, i.e., the courier, need not be a single-server station. In this case, the marginal probability distribution is given by

$$P(N_{R+1} = v) = \frac{r_{R+1}^{v} \cdot G(R, N-v)}{A_{R+1}(v) \cdot G(R+1, N)}.$$

If any of the other stations have multiple servers (more than one FS engineer), it is possible to obtain their marginal distributions. To do so requires permuting the indices of the stations in such a way that the one of interest becomes R+1, recomputing the matrix G and using the above equation.

The marginal distributions can be used to compute *average queue lengths* for each station. For single-server stations, it holds:

$$E[N_i] = \sum_{\nu=1}^{N} r_i^{\nu} \cdot \frac{G(R+1, N-\nu)}{G(R+1, N)},$$

and

$$E[N_{R+1}] = \sum_{\nu=1}^{N} \nu r_{R+1}^{\nu} \cdot \frac{G(R, N-\nu)}{A_{R+1}(\nu) \cdot G(R+1, N)}$$

It also holds:

 $E[number in queue of station i] = E[N_i]-u_i$,

and the *idleness* of a station i (i=1,...,R) is given by the probability

$$P[N_i = 0] = 1 - u_i,$$

and for the courier (station R+1):

$$P[N_{R+1} = 0] = \frac{G(R,N)}{G(R+1,N)}.$$

Now, concerning the *allocation of the FS engineers* to the customers population, this may be easily controlled by the FS manager, by finding the maximum N that satisfies the inequality:

$$E[N_{R+2}] = N - \sum_{i=1}^{R+1} E[N_i] \ge 0.95N,$$

when he/she wants to maintain a 95% level of service to his/her customers, which is an acceptable level for customer satisfaction.

References

- [1] Buzacott J.A. and Shanthikumar J.G. (1993) *Stochastic models of manufacturing systems*. Prentice Hall, New Jersey.
- [2] Buzen J.P. (1973) Computational algorithms for closed queueing networks with exponential servers. *Comm.* ACM, 16, 527-531.
- [3] Gershwin S.B. (1994) *Manufacturing systems engineering*. Prentice Hall, New Jersey.
- [4] Papadopoulos H.T., Heavey C. and Browne J. (1993) *Queueing Theory in Manufacturing Systems Analysis and Design.* Chapman and Hall, London.
- [5] Solberg J.J. (1976) Optimal design and control of computerized manufacturing systems. In *Proceedings* AIIE 1976 Systems Engineering Conference.
- [6] Solberg J.J. (1977) A mathematical model of computerized manufacturing systems. In *Proceedings 4th International Conference on Production Research, Tokyo.*

Picking Your Brains: A DSS for Neurosurgery

P. L. Powell*, N.A.D. Connell**, P. Lees#, and C.M.S. Sutcliffe**

*Information Systems Research Unit, Warwick Business School, University of Warwick, Coventry, CV4 7AL.

**Department of Accounting and Management Science, University of Southampton, Southampton SO17 1BJ.

> # Wessex Neurological Centre, Southampton Universities Hospital Trust, Southampton.

This research involves the construction of a decision support system (DSS) to assist clinical managers in costing and contracting in the new internal market for neurosurgery in the UK National Health Service. The research requires the use of a novel method to collect detailed patient costing data. This data is used to build up profiles of 'similar' patients for contracting purposes using clustering techniques. The clusters are employed in a simple DSS to evaluate and simulate the effects of cost and activity changes under different contracting scenarios. The detailed data allows further uses in clinical audit, outcome studies and macro-level health policy analysis.

Introduction

This paper illustrates how a clinical unit, the Wessex Neurological Centre (WNC) within the National Health Service (NHS) can cost its activities and contract with its purchasers in the new internal market. The research demonstrates a decision support system (DSS) constructed from data supplied by an innovative data capture method. This method allows clinical managers access to detailed activity and cost data which has uses beyond the current study. The research further investigates the worth of patient clustering categories widely used elsewhere. The following sections discuss the nature of the problem facing managers in the NHS and their data needs. Consideration is given to current methods for grouping patients and to the use of DSS in the medical environment. The process by which data is captured and used is discussed, along with sample data from one medical specialty - neurosciences. Finally, the applicability of this system to other specialties and of the data to other uses is considered.

Background - The Nature of the Problem

Health care costs are rising rapidly. According to the Economist (1994) "those who finance health care - i.e. governments and insurance companies - are responding by demanding more information about the cost, effectiveness and quality of the services they are buying". Against this background, in 1989, the UK government published a paper on the future of the NHS. One important thrust was to bring a new style of management to hospitals, enabling and encouraging the introduction of private-sector style attitudes and values. These would have a major impact on the way in which hospitals control and manage their resources. Greater financial accountability would be a first step to the creation of internal markets in which service agreements would be drawn up between District Health Authorities (DHAs) and fund-holding GPs (as purchasers of services) and hospitals (as providers). It became clear that the market philosophy would encourage the purchaser to buy services from whichever hospital was felt to give the "best value", and the providers similarly to sell their services to any DHA. To facilitate this, systems would need to be developed to ensure that costs and benefits are carefully and continuously monitored, and that the services offered are realistically, and perhaps competitively, priced.

The Contracting Process

Various types of contract have been proposed (under a general atmosphere of 'managed competition'), although not all have yet been fully implemented. The aim is to overcome problems inherent in the previous financial allocation system, namely that fixed money allocations to health authorities inhibited short-term increases in patient throughput volume, and also that since capital was essentially treated as a free good it was not efficiently managed (Ellwood, 1992). Although competition in health care has benefits, it is not without costs. As Ellwood points out, perfect competition requires, inter alia, consumers having a choice of suppliers, known prices and qualities and access to information. Estimates of the transactions costs of trading between purchasers and providers range from 10 to 20% of total costs and result in the need for regulation and control. In the internal market, money and patients go together so that any treatment should have payment attached.

Currently three types of contract are possible - block, cost and volume, and cost per case (figure 1). It is up to the purchasers and providers of health care to negotiate contracts on an annual basis. Block contracts specify a range of treatments to be carried out, the remuneration for which is not based on throughput; essentially the fee is for access. Cost and volume contracts have elements of the block contract, but above a certain specified throughput, treatments are charged on a cost per case basis. This additional charge is supposed to reflect the marginal cost of treatment if the extra throughput of patients represents the use of unplanned spare capacity.

This type of contract does, of course, necessitate more detailed costing data than block contracts. The final type, cost per case is costed on actual activity levels and requires the most detailed costing. The implications of changes in activity levels, and hence risks, for both purchaser and provider are clearly different for each contract type, and the contracts differ greatly in the volume and accuracy of data required to service them. Currently most contracts are of the block type but gradually both purchasers and providers are moving to cost and volume and cost per case. For example, the WNC has experienced a 20% increase in workload since 1989; such an increase would be difficult to sustain if all contracts were of block type. The price charged for any activity is supposed to be at actual cost, where cost includes depreciation and interest.

The internal market has been artificially constrained during the early years as a transitional measure to allow purchasers and providers to prepare. In particular, all the services offered by a particular hospital specialty are usually offered at a common price, rather than there being any attempt to set a price for each type of service. Such specialty-level pricing is crude; although most clinicians recognise that different patients incur different costs, most probably do not feel equipped to quantify differential costs and, even if they did, purchasers might be reluctant to accept them. There is, therefore, a widely felt need for a more accurate reflection of the different costs associated with treating various types of medical condition within any particular specialty. However, any contract which is not of a block nature will need to specify clearly the product or service being traded. Ellwood concludes that the direct care costs associated with a particular specialty are "of dubious credibility", and that the Department of Health would be ill-advised to remove the safeguard of the transitional restrictions on the market place, such as block pricing. This research offers a method for accurate cost compilation which will overcome some of these difficulties. Further, the transparency of the method enhances the acceptability of the results by all contracting parties.

Hospital Costing Data - deriving standard costs

Prior to the reforms, cost information was poor, indeed most data concerned highly aggregated activities and was not financial. The Economist suggests that "the writing and collection of patient records has changed little in response to the computer age, as doctors have been reluctant to abandon paper for electronics". What experience there is of cost collection suggested that most costs in the hospital sector are fixed or semi-fixed with, perhaps, three-quarters being staff costs (Ellwood, 1992). Coles (1989) suggests that up to 72% of hospital costs cannot be readily attributed to individual patients, though Coverdale et al, (1980) offer 60%. Under the new contracting system, hospitals will have to decide how to price the services provided by each specialty. The form of these new contracts is largely under the control of the purchaser and provider. Indeed, many purchasers are

seeking a charge for each finished consultant episode (FCE), and so the relevant DHA needs to set a price for each `patient episode' treated.

An episode begins when a patient is admitted, and continues until they are discharged from the care of a particular consultant - hence a single admission may be made up of a number of FCEs. Since every patient episode is unique, costing and pricing could be achieved using a `job costing' approach, in which the hospital records the actual cost of providing the service received by each patient. However, this would require a large continuous data collection exercise, and a preferable alternative is the establishment of `standard costs'. Indeed, the former approach is contrary to the views of the NHS Management Executive (NHSME) and is resisted by most purchasers. In the standard costing case, each patient episode is assigned to a pre-defined group that typically requires a similar amount of resource to treat. It relies on the ability of the provider's management to define cohesive treatment groups and to estimate the resources likely to be used in providing a service for each. However, as in any standard costing approach, variances are likely due to changes in input prices, volumes, case-mix, efficiency and treatment patterns (Ellwood, 1992).

Previous Costing Attempts

In 1984 the WNC admitted 1185 patients with a mean stay of 12.5 days at a mean cost of about £2500 per operated patient and a cost per occupied bed day of about £146 (at 86% occupancy). It is interesting to contrast this with the 1992 Department of Health figures which suggest that the cost of the average patient week was £1,072 up from £991 in 1991. Not only are costs of care rising rapidly but specialisms such as neurosurgery are more expensive than general medicine.

The first problem involves disentangling the cost of operating the Unit from the costs of operating the remainder of the hospital. This is a daunting tasks since the activities of the Unit and the hospital are complex and inter-related. For example, the hospital is a teaching one, there are centrally funded pathology services, there are other regionally funded services and the site also contains residential accommodation for staff. Further, the accounting system did not distinguish between costs incurred by the unit and those incurred elsewhere; and the functional rather than cost centred organisational framework precluded pricing for a lot of services. Overhead apportionment was often arbitrary.

The essence of the costing problem is that, often particularly at micro-levels such as the patient, the resources used in the production of the service cannot be identified unambiguously. Usually resources are used by many patients and so the costs are joint. Overhead costs present additional problems. In some cases the supplying unit offers a price for the service (for instance all pathology tests of a certain type are priced similarly). Alternatively average costs may be used or actual costs based on the quantification of all resources used. Further difficulties in costing arise when there is not a single price for any resource. For example, nurses on different grades cost different amounts, just because an 'F' grade nurse carries out an activity for a particular patient does not mean that an 'A' grade could not or will not have done the task. Even in non-staff costs, differences may appear if the same goods are purchased in bulk or individually. This becomes more of an issue in the internal market if a unit standing between purchasers of services on one hand and being purchaser of services on the other has different types of contract with each. For example, the WNC may have a block contract with one DHA but have to pay for each individual pathology test. Such individual costs will not be represented fully in the contract price paid. Perhaps the most important distinction is between variable and fixed costs. Costs are fixed or variable in relation to a measure of output, so the distinction helps elucidate the cost implications of changing input levels. Costs are only non-variable within a particular time and over particular ranges of output. Many resource use costs have both a fixed and a variable element.

This work addresses the issues highlighted. The costing problems outlined exist and the introduction of the internal market has given fresh emphasis to the activity cost issue. On the benefits side, technology has improved and functional reorganisation of the hospital and budgetary devolution to the WNC has assisted in drawing boundaries around the unit. However, the nub of the problem, identifying, classifying and recording activities, durations, uses and costs remain. It is against this background that the current research attempts to shed light, having developed a new method for so doing.

Diagnostic Related Groups

Many attempts have been made to derive patients groups and thus standard costs. The earliest proposals were clinically-based, trying to identify homogeneous groups of patients who had similar clinical properties, for example, the same illness or injuries. Whilst progress has been made in this type of clinical analysis, little attention has been paid to the management properties of patient groups. Thus, although patients might have similar clinical problems, there is no suggestion that the costs of treatment might be similar: no two broken legs or head injuries are identical. The lack of cost data persisted until the US Government investigated hospital reimbursement schemes based on patient types and standard costs. One way in which standard costs can be achieved which has aroused much interest in the US and has been evaluated in over 20 countries is the use of diagnostic related groups (DRGs); patients with similar medical conditions are grouped together, and costs associated with each group constructed.

According to Fetter (1991), one of the originators, DRGs were designed to allow hospital performance to be measured and evaluated. In essence, the developers

searched for the "simplest regulatory mechanism that could substitute for the absence of an open market in healthcare". This market was conceived never to be fully open since health consumers have little information about the value and quality of treatment and, by and large, do not pay for most services. The mechanism for DRG use (Scarpaci, 1988) is to, first, assign the patient to one of 23 major diagnostic categories. Then the hospital payment rate per case is multiplied by the weighting of the assigned DRG. The weight represents the mean resources expended on a medical/surgical procedure relative to the mean national amount of resources consumed per case by an average hospital. The patient is then assigned to one of more than 468 DRGs (the number has slowly increased as some are subdivided in the search for greater homogeneity). As Scarpaci states "the assignment of a patient to one of 468 DRGs does not represent a definite diagnosis, a total account of itemised charges nor a comprehensive evaluation of patient care. Rather DRGs are a tracking facility that enable hospitals to operate as multi-product firms whose outputs are medical and surgical procedures. Instead of using the traditional measures of hospital performance such as the proportion of medical staff specialists, bed size and occupancy rates, DRGs gauge performance based on treatment procedures and patient attributes".

In the neurological area there are only a small number of DRGs - 36 - divided between, surgical and medical principal diagnoses. For example, in the surgical group, patients would be assigned to a different DRG based on whether the principal diagnosis of their problem was intercranial, spinal, extracranial vascular, or peripheral and cranial nerve. If the condition is medical, examples are brain tumours, degenerative nervous system disorders, multiple sclerosis, and head injuries.

Problems with DRGs

Although DRGs are a step in the right direction, it is clear that there are a number of difficulties which might restrict their use for costing and contracting. For example, Greenhalgh and Todd (Bevan, 1989) question the use of DRGs, which rely "on the assumption that those in the same diagnostic group consume the same amount of health care resources [which] implies that patients are treated according to their diagnoses regardless of the individual clinical characteristics or treatments". Bevan suggests that homogeneity is only meaningful over a large number of patients not at an individual level. It may be that the vital decision in terms of resource use is the doctor's decision to admit the patient rather than factors such as length of stay. This concurs with Babson (1973) who found that over 90% of total direct patient costs are incurred in the first 30% of any stay, and Sanderson et al. who state that the last few days of any stay tend to be cheap. In summary, Fetter (1991) suggests that there are three main problems in predicting costs of services; not all diseases are equally well understood, treatments for the same disease differ, and the coding of illnesses is difficult.

It should be clear that for any `standard cost' approach to be useful, clinicians and managers need to have confidence in the groupings. This needs to be in terms of the extent to which the costs associated with each group are an accurate reflection of actual costs and therefore a sound basis for pricing. Williams et al. (1982) find that average costs do not reflect well the actual use of resources. Horn and Schumacher (1979) contend that average cost is not useful since for most DRGs the standard deviation of cost is greater than the mean. Manton and Vertrees (1984) point to the instability of average cost within DRGs, since more categories implies fewer patients in each and also because the high costs during the terminal stages of chronic illnesses of the elderly are not reflected in DRGs. The requirements for any groupings are that they should be comprehensive and mutually exclusive, clinically coherent and homogeneous in their use of resources. Rosko (1988) goes further suggesting that any good patient classification system needs to have certain characteristics, viz. be based on reliable data and on iso-resource use groups of patients, there needs to be a manageable number of categories, the system needs to be meaningful to doctors, and the information has to be accessible at reasonable cost.

In an attempt to overcome some difficulties, additional factors are reflected in some There is considerable debate as to whether age or comorbidity is DRGs. preferable. Comorbidity is present when the patient has more than one disease or injury though the secondary ones may be minor compared to that used for DRG classification. The developers of DRGs investigated comorbidity but found that "most of our hypotheses about what ought to make a difference could not be supported by the data...when all else failed we used age" (Fetter, 1991). Subsequent work, however, pointed to the dominance of comorbidity over age as a differentiating factor, and so comorbidity has now replaced age in 95 DRGs which had age categories. Yet, only half of the 18 neurosurgery DRGs used here are stratified by the presence of complications, although Munoz et al. (1988) assess complications and comorbidities as directly influencing both costs and length of stay, while age did not. Again, Desharnais et al. (1988) found that age alone, in the absence of comorbidities does not result in substantially higher costs or longer stays. They find that patients over 70 may have an average length of stay of about half a day longer that those under 70, whereas those with comorbidities have stays three days longer. Perhaps the most striking finding here is that even patients aged over 90 without comorbidities were discharged faster than younger patients with comorbidities. Age, according to Jencks and Dobson (1987), is a poor predictor of cost but a strong predictor of death.

Alternatives to DRGs

The doubts as to whether DRGs are the most effective indicator of costs have led to the development of healthcare resource groups (HRGs) in the UK. HRGs, modified DRGs, are seen to reflect more accurately UK clinical practice and cost profiles; similar modification of DRGs has been undertaken in the Netherlands (Verheyen and Nederstigt, 1992). In the UK, since cost data is limited, length of stay is widely used as a proxy for resource use. Even though most hospital costs are fixed, this, coupled with imperfect diagnostic coding, restricts the accuracy and development of HRGs. As discussed, the problems with DRGs and similar measures are that they do not take into account severity, nor length of stay. Indeed few DRGs are homogeneous across acute specialities. In fact, they may just reflect faulty clinical decisions or faulty execution (Rosko, 1988). Fetter (1991) gives examples of two DRGs for which resource utilisation may vary by a factor of seven between hospitals and a factor of ten between cases. Johansen (1986) finds that DRGs are inferior to other methods in explaining variability in length of stay. Sanderson (1989) argues that some DRGs are homogeneous with respect to length of stay whilst others are not. McNeil et al. (1988) find some DRGs can be improved in terms of homogeneity whilst others cannot, and it is not always clear how this may be done clinically. Similarly, Horn and Schumacher (1982) maintain DGRs are the least adequate method for patient grouping when compared using two statistical criteria for homogeneity.

McMahon and Newbold (1986) find physician practice to explain more intra-DRG variability than does severity. There are also other factors which may affect intensity of care such as emergency versus planned admission and clinical decisions such as the choice to treat for survival or for pain relief (Jencks and Dobson, 1987). They highlight two further issues. First, patients considered outliers in terms of DRG category are significantly more likely to have been outliers before in previous episodes, and, second, certain doctors always had a higher proportion of patients classified as outliers compared to their colleagues. Initial work by Greenhalgh and Todd (1985) points to "marked differences in the use of resources per day between patients with exactly the same diagnosis and between patients requiring the same treatment procedure". In similar vein, Horn et al. (1986) state "it is now well accepted that individual DRGs are not homogeneous with respect to resource use". Berki (1983) goes further " there is as yet no taxonomy based on clinical data that can demonstrably classify cases in terms of their severity and clinical case management complexity into a set of exhaustive and mutually exclusive homogeneous cells". Berki sees DRGs and disease staging as currently the best developed, though neither yields a set of iso-resource categories within acceptable effectiveness criteria. A further candidate criteria for differentiation is outcome. Little work has been performed relating outcomes to costs, however, Munoz et al. (1989) calculate that mean total costs for patients who die are over five times those for patients who survive. Similarly, the former have an average length of stay over four times greater than survivors.

It is clear that the WNC needs to have a basis for costing and contracting. Yet, following from the above, the question is, if DRGs or HRGs are not a sufficiently reliable ingredient in the pricing model, are there more representative groupings which can be used? There is a clear need to assess current classification possibilities and to evaluate their effectiveness for costing and contracting.

Testing DRGs, HRGs and Standard Costs

The WNC faces the problems described above. As a budget-holder, the WNC recognises the need to manage its resources, and acknowledges that one way this might be achieved is to measure costs accurately. It is an acute unit, dealing with disorders of the spine (degeneration/disc disease, tumour) and brain (tumour, trauma, haemorrhage) many of which are emergencies and, therefore, currently provided as part of a block contract. The WNC is a self-contained unit offering a spectrum of treatment ranging from 'cold' investigation to neurological intensive care, and it has already suffered financial problems due to major shifts in case-mix within the block contracts (British Medical Journal, 1992). Block contracts have emphasised the need to understand the effects of case-mix changes and its resource consequences.

The WNC has three wards, including an intensive care unit. Managers are already aware of fixed costs, and see ways in which some variable costs, for example costs arising from drugs and tests, could be captured and analysed. However, one of the most significant costs is staff, which is semi-fixed, and it is difficult to relate costs precisely to particular types of patient episode or categories of activity. The project recognises the need for a more accurate pricing model based on a better understanding of variable and semi-fixed costs. Hence, the model forms part of a DSS which could be equally useful to other specialisms in preparing bids to purchasers. Further, it could also be used by purchasers to experiment with the effects of demand changes. As identified, the first task is to consider alternative ways in which patient episodes might be grouped, and to implement a method by which the costs of service provision, particularly staff time, could be more accurately measured.

Grouping of Data

Greenhalgh and Todd (1985) identify lack of available data as inhibiting the effective financial management of clinical units. They maintain that accurate patient costing requires identification of the activity which consumes resources in terms of staff time and consumables. Ashford and Butts (1979) add, "all that is required to relate the financial inputs to the patient care outputs is to compile a

detailed record of all types of services provided". However, they go on to state that this cannot be done directly. Yet, technology has advanced considerably since 1979. Manton and Vertrees (1984) suggest that in addition to DRGs, other data might be useful including discharge status, auxiliary diagnosis, age and sex, and treatment. A large quantity of this data about each patient is already collected in the WNC. This takes a variety of forms, and is collected for a variety of purposes; some is demographic, some historical, and only some is current.

The `output' of the WNC is defined in terms of FCEs; a patient enters the service, moves through it (consuming resources), and is discharged (perhaps to a different specialty or perhaps to the care of a GP). Each episode generates various activities, (nursing, intensive nursing, surgery, admission, discharge) and costs can be calculated for these and attached to that episode. If there is a satisfactory match between types of episode and DRGs or HRGs, such that managers can be confident that the latter form a reasonably accurate aggregation of episode types, it is necessary only to gather costs.

Within any particular specialism, such as neurosurgery, there might be quite a modest number of HRGs. If, however, there is not a satisfactory match between HRGs and costs associated with particular patient episodes, then a different grouping might yield a more accurate determinant of costs to be fed into a pricing model. This might involve modification of the HRGs or developing different groupings. Miles et al. (1976) were some of the first to attempt to group or cluster patients and claim that if customer classes can be measured by attributes, then one has the basis for control of the 'production' process. This control is of cost, via classification of patients according to their patterns of resource consumption, with the caveat that the classes must be medically meaningful. One aspect of the current research project is to undertake a comprehensive analysis of the determinants of variable cost for a range of patient episodes, to determine groupings that produce data which is more useful for contracting. Ultimately, however, the final determinant of the grouping method chosen must be its usefulness to clinicians and managers.

Accurate Measurement of Activity Costs

The above requires an accurate method of capturing data. Mindful of Coles' (1989) statement that "patient-based costing would require a marked increase in information gathering and computerisation", and Doremus and Michenzi's (1983) findings of substantial errors in hospital discharge data, two factors were influential in determining an appropriate method. First, staff are already required to record a variety of data, and it is important that costing data is seen as complementary rather than duplicating existing collection. Where possible, any proposed system should either draw upon existing data sources, or provide an alternative, more acceptable, way of meeting requirements. Second, health professionals are already operating

in a demanding environment and it is therefore important that any system to monitor and record activities should not place an additional burden upon them, and have a perceived operational benefit. An obtrusive data collection system may also be perceived negatively by patients.

The initial phase concentrated on establishing the chronological flow of patients through the WNC, identifying key activities. From this, the complete set of activities undertaken could be established as a precursor to measuring activity durations. Clearly, the first task in analysing activities is to record all activities undertaken. A number of possible methods exist for this, however here the interest is patient-centred, so the chronological flow of a patient is used as the recording process, and the work clarified, probably for the first time, the range and nature of the activities carried out, as a necessary precursor to data collection.

Data Capture and Decision Support

In the light of the above constraints, a variety of data capture methods were investigated. Past hospital data collection methods have relied on cumbersome systems such as form filling or retrospective keyboarding by nurses and doctors. None has the desired characteristics and most have been rejected by those who operate them. There is, however, a method, common in retail business which has potential here. The project team decided to barcode all activities, and to use portable light pens to capture the data.

Each activity (for example, bed bath or giving drugs) is given a unique code printed on laminated pads, either carried or strategically placed, to be scanned during the performance of the activity. In addition, each patient has a unique code, generated on admission, which acts as a label to which activities are attached. Each activity performer (nurse, doctor, physiotherapist) is given their own, uniquely coded, light pen. The pens are compact, a little larger than a standard ink pen. Each pen is equipped to capture the start and finish times of each activity and at the end of each period is downloaded to a central PC. Note that is important to capture not only that an activity took place but the duration and also the type (or grade) of staff who undertook it since users may be of different types or grades, and therefore represent different costs.

In order to ensure completeness in the recording and to eliminate the possibility of unaccounted down-time, a hierarchy of bar codes is used. This starts at the level of the medical staff - doctor, nurse, physiotherapist etc., the second level is the patient upon whom the activity is performed, the third level concerns the actual activity, while the lowest level records the duration. Similar hierarchies are used for recording drug usage and operating theatre activities. The approach is patient-centred since the interest is to distinguish different activity levels for different patient classes. Figure 2 illustrates the range of activities covered by the exercise

for both direct and indirect resources. Although there may be recursions, most patients will conform to the chronological flow outlined in figure 3. This is used to identify activities occurring at any stage of the patient's progress. Thus it begins with the clerical activity of admission and follows a patient through pre-operative nursing, medical and investigation activities to surgery and recovery. Due to the nature of the specialism, most patients require intensive post-operative care before they are moved to a lower level care ward and then discharged.

Although used in a variety of retail and manufacturing environments, bar-coding has not been widely used in health care. Despite the first mention in 1986 (Rappoport, 1986), it has not found general use and, where it is used, is confined to activities such as laboratory use. Timpka et al. (1992) find two major health-care uses - stock and item distribution and clinical laboratory applications. The authors see the benefits as increasing ease and speed of data input and increasing accuracy. Though the former is sometimes disputed, the latter is generally true. The main problems is a lack of standardisation and difficulty in integrating resultant data into other systems. Both these difficulties did not pose a problem here, although some software had to be specially written. Interestingly, Timpka et al. do not envisage any role for bar-coding in health-care planning and administration, though they state there is potential in clinical audit work. Figure 4 illustrates some of the barcodes developed in the project and their associated hierarchy. In similar vein, figure 5 details the data collection activities undertaken in the operating theatre. This demonstrates the multiple participants and multiple simultaneous activities which may be taking place during a patient episode.

Each patient's episode of care is analysed in terms of the medical interventions, the activities these generate, and the associated costs. A picture emerges of cost drivers for particular patient groups, which can be fed into a pricing model to allow, for instance, sensitivity analysis on particular groups for contracting purposes. Even crude analysis of the data has enabled preliminary observations about standard timings for commonly-performed activities. The end result of the costing phase is the processing of the data to produce clusters of patients. The clusters are of `similar' patients, that is, patients who consume similar amounts of resources. Similarity may depend on illness, length of stay, age or other attribute. When manipulated into manageable groups, these form the basis of contracting and of the DSS. The DSS will allow managers to simulate the effects on activity and costs of various contract types and different bases for contracting. This should enable better planning and more efficient resource utilisation.

Figures 6. and 7. show details of data prior to expert filtering (to remove outliers) and clustering. Filtering is necessary to remove obviously implausible data. For every activity, nurses and doctors who regularly perform it were questioned as to the maximum and minimum possible value the variable might take. In some cases these were augmented by trial timings of activities. For instance, there is a

minimum time possible for taking the patient from the ward to the operating theatre. This involves use of a lift which may not be readily available, and a distance to travel. Similarly, to get maximum potential values, an informal risk analysis was carried out. Staff were asked to imagine a worst case scenario, where everything possible might go wrong for the most difficult type of patient. Note that patient "difficulty" is not necessarily the same as severity. A severely ill patient may be in a coma and therefore passive, a less ill patient may resist. Similarly, mobile patients may be able to perform some tasks for themselves - this may not be the quickest way to carry out tasks but may be part of rehabilitation. This process of filtering provided an upper and lower bound for the data. After timings were obtained, data which lay outside the bounds were candidates for removal. However, data was fed back to the experts if a significant number of readings exceeded the bounds. This process is vital since the data is used to differentiate patients, and so data was not excluded unless there was complete agreement that it was erroneous.

Figure 6. demonstrates the range of activities performed on individual patients. The most intensive count of activities on a particular patient, 1828 activities, during his/her FCE is nearly twice that of the fifth patient and compares to a mean number of significant activities per day of around 30. The median patient in the data collection exercise had fewer than 200 activities performed. Clearly, vastly different resource usages are implied by this activity data, yet many of these patients would be classified in identical DRG categories. Any contracting process would need to differentiate such patients. Figure 7 details the mean and standard deviation durations of some activities. Again, the very large standard deviations imply that different patients consume non-similar resources. Figure 8 presents the data graphically, demonstrating the shape of the distributions. Initial analysis by the project team has focused upon "dependency scores" as a possible surrogate for resource use, although further work is required to establish more precisely the strength of this. The clustering process takes the bar-code data in order to produce groups of similar patients. The resultant clusters provide the input to a simple robust DSS used by staff as a basis for contracting.

In the costing and contracting problem here, the data derived from the bar-coding system provides the data-base of the DSS, the model-base is the set of tools used to investigate the behaviour of costs under various scenarios and contracting possibilities. Because the DSS needs to be readily available and usable by staff, a standard DSS generator, a spreadsheet, is used to display results and allow user interaction. The system developed has aspects of both institutional and ad hoc DSS. It is necessary to automate as much of the data analysis as possible to provide management with standard outputs on a regular basis. This facilitates control. However, for annual contracting negotiations, users will want to experiment and simulate alternatives. A standard DSS generator is used because it

needs to be compatible with current systems and expertise. As shown, the problem solution is data intensive. That is, obtaining and processing the data for the DSS posed the bulk of the work. Once the costing data is available and clustering has taken place, the DSS is relatively straight forward.

Conclusion and Future Potential

Although the data produced by the system is primarily collected for contracting, it will have a multitude of uses elsewhere. For example, figures 9. and 10. show early results of analysis of nurse time by grade. It can be seen that the preparation of shift report is highly labour-intensive while full neurological observation is similarly time-consuming. This highlights potential areas for management or technological intervention. Given that the data is categorised by grade, management can assess whether the appropriate grade of nurse is undertaking each task. Conversely, when looking to support the Unit, consideration can be given to whether investment in appropriate technology can improve efficiency or effectiveness in those tasks which take up considerable nurse time.

In terms of using clustered data, Donaldson and Magnussen (1992) feel "it is doubtful whether the use of DRGs...can deliver more efficient clinical practice" In addition to testing DRGs and other measures this data (consider, for instance, changes in severity during a patient episode) may have a role in clinical audit, in looking at the outcome effects of different treatment profiles and in wider studies on issues such as the effects of ageing on the population. This involves the comparison of the use of resources by different age categories and projection of the results as the average population age increases. Not only is the population ageing but there is an increase in the very old. The proportion of retired people in the UK will increase from 18.4% to 19.8% by the next decade. Based on present consumption, health care costs are expected to rise by 40% by the year 2025. There are also impacts from improved health which compresses morbidity into a smaller time-frame, from advancement of medical technology resulting in greater opportunities for health improvement but also raised expectations. This is compounded by increased education of patients - better informed perhaps meaning more demanding (Lagergren, 1993). Although total heath care volumes and hence costs are rising rapidly, disaggregating the total shows that acute hospital care is outstripping other components - chronic, psychiatric and primary. Thus, data from this study might give an important insight into likely future health care costs especially in acute units. Lagergren suggests that the only solutions to restrict rising costs are to structurally redesign the health system (reduction in hospital beds, day-surgery, improved co-ordination between health authorities and social services), increasing efficiency by managed competition and patient choice, improved systems for management, cost control, monitoring and evaluation and lastly, more accurate assessment of needs, outcomes and quality of care. It can be

seen that this project provides support for some of these. The Economist identifies the crux of the problem of clinical audit and outcomes, "if health-care systems are to be made more efficient there must be some way of measuring their input (sick patients) and output (cured ones)". In clinical audit, the system, because it is patient centred, gives an audit trail of what happened to any given patient and when. In the investigation of specific outcomes profiles of similar patients can be investigated to highlight if practice differed significantly between patients. This may provide a valuable learning tool for clinicians. In a study of outcomes of neurological care at the WNC Pickard et al (1988) suggest that neurosurgical procedures appear to offer relatively good value for money. However, they point to the real differences in the cost effectiveness of treatments offered by neurosurgeons, implying that certain procedures are more beneficial than others. They report that monitoring trends in cost effectiveness would be useful in determining whether services are evolving appropriately, but recognise that such monitoring requires greater accuracy and consistency in both the cost and the product of care estimates. This necessitates considerable work in allocating costs to individual patients - which the current system provides. Lastly, they highlight the need to see the cost effectiveness of a new treatment in the context of the overall cost effectiveness of managing a patient to final outcome rather than simply in terms of the immediate burden on the budget. Perhaps the acquisition of cost effectiveness data can be used to demonstrate to purchasers the need to consider total, long term costs when engaging in the contracting process.

A further use of the data is to develop cost drivers for the Unit. While patient throughput is undoubtedly a major consideration, there is evidence (Thorpe, 1988) that some costs are related more to staffed beds than output and it may be that the ratio of forecast to actual admissions is more important than actual admissions. Thorpe finds that cost differences are more related to teaching activities than to any other cause, while case-mix, wage-rates and service intensity are also significant. Complexity and length of stay directly influence costs while specialisation does not (Barer, 1982). As discussed the decision to admit is a key cost driver with hospitals showing considerable variation in the incidence of hospitalisation for similar diagnoses (Wennberg et al., 1984). Hence, the activities of individual physicians is of interest. Having activity data on individual patients should allow professional discretion to be highlighted.

Lastly, the WNC has to follow the directives of the NHSME who consider it to be performing well if it is meeting Patients' Charter standards, providing publicly stated evidence of outcomes and meeting financial returns of 6% p.a.. The NHSME recognise that current performance is variable and that information systems are poor. They see the internal market as a means to an end, driven by common sense not ideology and emphasising dialogue between purchasers and providers.

This work has highlighted the effects of the new NHS internal market on one specialism, neurosurgery. It has shown the need for accurate, timely and relevant data to guide the activities of those involved in the contracting process. It has demonstrated a robust, yet inconspicuous, method of data capture and the mechanism by which this might be implemented. It reports the results of the data collection exercise which clearly demonstrate the need to base contracting on resource usage. Finally, it indicates how this data is be used to provide support to clinical and management staff by the provision of a DSS. While previous work has pointed to a lack of alteration in clinician behaviour after the provision of cost data (Wickens et al., 1983), this work has been driven by clinicians who recognise that they must now take resources into account. The data collection system can also be used routinely for planning and control and could easily be adapted for other specialisms since the methodology is readily transferable.

References

Babson J., Disease Costing, Studies in Social Administration, Manchester University Press, 1973.

Berki S., The Design of Case-based Hospital Payment Systems, Medical Care, vol. 21 no. 1, pp. 1-13, Jan. 1983.

Barer M., Case-mix Adjustment in Hospital Cost Analysis, Journal of Health Economics, vol. 1, pp53-80, 1982.

British Medical Journal, 1992.

Coles J., Attributing Costs and Resource Use to Case Types in Bardsley M., Coles and Jenkins L., (Eds), Kings Fund, 1989.

Coverdale I., Gibbs R. and Nune K., A Hospital Cost Model for Policy Analysis, Journal of the Operational Research Society, vol. 31, no. 9, pp.801-811, 1980.

Desharnais S., Chesney J. and Fleming S., Should DRG Assignment be Based on Age?, Medical Care, vol. 26, no. 2, pp124-131, Feb. 1988.

Doremus H. and Michenzi E., Data Quality, Medical Care, vol. 21, no. 10, pp.1001-1011, Oct. 1983.

Donaldson C. and Magnussen J., DRGs: The Road to Hospital Efficiency, Health Policy, vol. 21, no. 1, pp.47-64, 1992.

Economist, The Future of Medicine -a Survey, 19 March 1994.

Ellwood, S. Cost Methods for NHS Healthcare Contracts, CIMA, 1992.

Fetter R., DRGs: Understanding Hospital Performance, Interfaces, vol. 21, no. 1, pp.6-26, Jan 1991.

Greenhalgh C. and Todd J., Financial Information Project: Message for the NHS, Brit. Medical Journal, vol. 290, pp.410-411, Feb. 1985.

Horn S., Horn R., Sharkey P. and Chambers A., Severity of Illnesses within DRGs, Medical Care, vol. 24, no. 3, pp.225-235, March 1986.

Horn S. and Schumacher D., An Analysis of Case-mix Complexity Using Information Theory and DRGs, Medical Care, vol. 17, no. 4, pp.382-389, April 1979.

Horn S. and Schumacher D, Comparing Classification Methods, Medical Care, vol. 20, no. 5, pp. 489-500, May 1982.

Jencks S. and Dobson A., Refining Case-mix Adjustment, New England Journal of Medicine, vol. 307, no. 11, pp.679-686, 1987.

Lagergren M., Future Demand and Supply of Health Services, ORAHS Conference, Brighton, July, 1993.

Manton K. and Ventres J., The Use of Grade of Membership Analysis to Evaluate and Modify DRGs, Medical Care, vol.22, no. 12, pp.1067-1082, 1984.

McMahon L. and Newbold R., Variations in Resource Use within DRGs, Medical Care, vol. 24 no. 5, pp.388-397, May 1986.

McNeil B., Kominski G. and Williams-Ashman A., Modified DRGs as Evidence for Variability in Patient Severity, Medical Care, vol. 26, no. 1, pp.53-61, Jan. 1988.

Miles R., Fetter R., Riedel D. and Averill R., *AUTOGRP: An Interactive Computer System for the Analysis of Health Care Data*, Medical Care, vol. 7., pp.603-615, 1976.

Munoz E., Chalfin D., Birnbaum E., Mulloy K., Johnson H. and Wise L., Hospital Costs, Resource Characteristics and the Dynamics of Death for Surgical Patients, Hospital and Health Services Administration, vol. 43, no. 1, pp.71-83, Spring 1989.

Pickard J., Bailey S., Sanderson H., and Rees M., Steps towards Cost-Benefit Analysis of Regional Neurosurgical Care, 1988.

Rappoport A., *The Use of Machine Readable Patient and Specimen Identification to Enhance Clinical Laboratory Quality Assurance*, Informatics in Pathology, vol. 1, pp. 1-5, 1986.

Rosko M., DRG and Severity of Illness Measures: An Analysis of Patient Classification Systems, Journal of Medical Systems, vol. 12, no. 4, pp.257-274, 1988.

Sanderson H., Storey A., Morris D., McNay R., Robson M. and Loeb J., Evaluation of DRGs in the NHS, Community Medicine, vol. 11, no. 4., pp.269-278, 1989.

Scarpaci J. DRG Calculation and Utilization Patterns: A Review of Method and Policy, Soc. Sci. Medicine, vol. 26, no. 1, pp.111-117, 1988.

Sharkey P., DeHaemer M., Simmons L. and Horn S., Assessing the Severity of Patients' Illnesses to Better Manage Health Care Resources, Interfaces, vol. 23, no. 4, pp.12-20, July 1993.

Timpka T., Nyce J., Sjoberg C., Renvall H. and Herbert I., *Bar Code Technology in Health Care: Using a Business Model for Study of Technology Application and Dissemination*, Proceedings of the 10th Annual European Federation for Medical Informatics Conference, Freund, pp.767-772, 1992.

Verheyen P. and Nederstigt P., A Cost-allocation System Applied to Dutch Hospitals, European Journal of Operational Research, vol. 58, pp.393-403, 1992.

Wennberg J., McPherson K. and Caper P., Will Payment Based on DRGs Control Hospital Costs?, New England Journal of Medicine, vol. 311, no. 5, pp.295-300, August 1984.

Wickens I, Cole J., Flux R. and Howard L., Review of Clinical Budgeting and Costing Experiments, British Medical Journal, vol. 286, pp.575-578, Feb 1983.

Williams S., Finkler S., Murphy C. and Eisenberg J., Improved Cost Allocation in Case-mix Accounting, Medical Care, vol. 20, no. 5, pp.450-459, May 1982.
WESSEX NEUROLOGICAL CENTRE

PATIENT CENTERED ACTIVITY STUDY

DIRECT RESOURCE INPUT



Figure 1. Patient Centred Data Collection



CHRONOLOGICAL PATIENT FLOW

Figure 2 Chronological Patient Flow

Figure 3 Bar Code Examples

Figure 4 Data Collection Example

Figure 5 Activity Count per Patient

Figure 6 Activity Duration - Means and Standard Deviations

Figure 7 Sample Activity Durations and Frequencies

Picking Your Brains: A DSS for Neurosurgery

II. Applications

A Designer's Decision Aiding System: DDAS

Jenny Darzentas, Thomas Spyrou, Eftihia Benaki, John Darzentas

Research Laboratory of Samos, Dept. of Mathematics, University of the Aegean

This paper describes the development of a working system for aiding designers of computer systems find appropriate tools and methods to enable them tackle usability problems. The approach taken for the design of the system is based on Soft Systems Methodology (SSM) and fuzzy reasoning in the form of Test Score Semantics and has been extensively described elsewhere. Here the building of the knowledge base and features of interaction with system are explained and illustrated with an example. The importance of the DDAS, lies not just in its usefulness to the designer, who now has access to bodies of knowledge in direct relation to a problem of concern, but in its claim to provide a methodology for decision aid in similar situations where problems exist, and tools to solve them also, but where a short cut or aid is needed to bring the two together

Keywords: decision aiding systems, design, human computer interaction, expert systems.

Acknowledgement: The work reported on in this paper was funded by the ESPRIT Basic Research Action 7040 AMODEUS (Assaying Means of Design Expressions for Users and Systems)

1. Introduction

The work described in this paper was carried out as part of the Amodeus project, one of the world's largest multidisciplinary HCI consortiums, developing modelling and analytical techniques for Human Computer Interaction.

The motivation for the genesis of the DDAS was as a transfer activity i.e. informing design practitioners of the potential of the Amodeus techniques. The approach that was taken to study the problem situation and design the system has been extensively described elsewhere [5,6,7,8].

The starting point for the architecture and development of the system is the assumption that the design practitioner confronted with a usability problem, - a problem which may be well articulated or only vaguely suspected, - would need some help to assess which techniques are best suited to solve this problem. In effect, by a designer decision aiding system is meant a system whose purpose is to

help a designer by first solving the problem of finding the most appropriate tools, within a specific array, to tackle the situation of concern.

In developing the DDAS some immediate limitations had to be imposed upon this scenario, in order to define the scope of the system.

Firstly and most obviously, it is not possible to offer aid for all problems or classes of problems, but only those that the Amodeus modelling techniques can handle. However, the array of Amodeus techniques covers design from a user oriented, task oriented, system oriented and design rationale points of view, and while one cannot claim that they cover all possible problems, it would probably be true to say that at least at some level of granularity they touch upon all.

Secondly, it is not possible to allow the designer-user to express his problem freely in natural language, because all effort would go to processing the input and trying to match it to the knowledge base of problems handled by Amodeus. Nor would this state of affairs be very desirable. Recent work in active DSS show that the user prefers "active" aid from a system, and wants to be prompted [11]. A user who has a very well formed idea of what his problem is would not mind expressing it , (though he may wonder if the machine is interpreting it as he wants), but a user who just "has a feeling" would have difficulties making that intuitive response to a situation comprehensible to the system.

Thirdly, the DDAS is designed at present to do no more than assess the user's need for technique(s) and make a recommendation and present this to him with some justification which will show the reasoning the system used to arrive at its conclusion. The system does not store knowledge about how a technique is used, or what skills are needed to use it. Thus it could be entirely possible that a recommendation is made which requires a background in software engineering which the user has not been assessed for. The reason for separating out the "what it does" from the "how it does it" was to remain faithful to a desire to have the techniques "compete" on similar terms and eliminate as much as possible constraints that would impose a too early limiting of choices upon the user. From a practical point of view as well, if a problem could be aided by a technique which requires, for instance, skilled personnel, then it is not unlikely that the design practitioner will take steps to acquire the services of said personnel, if at all possible.

The ways for dealing with the above three limitations were, firstly, to build a knowledge base containing descriptions of the aspects of the overall design space which are relevant to the modelling techniques; secondly, to present these descriptions to the user and ask him to select those which most closely resemble his own problem of concern; and lastly, have DDAS complemented by a system, or even documentation, such as the Executive Summaries [2], or a combination of multimedia documents describing a technique, showing it in action, etc. that would

cover supplementary information such as what it is, how it does it, who can do it, what is needed, how to interpret the results, etc. The approach to eliciting the knowledge and designing the overall system was based on SSM [3,4] and has been presented elsewhere [7].

The next section presents details of the implementation of the system, in terms of architecture, and technical details; building and organising the knowledge base, and features of the interaction. In section 3 an example session is given to illustrate the interaction, and section 4 presents conclusions and discussion.

2. Implementation

2.1 Architecture and technical details

The system architecture is represented in Figure 1. This shows the three main components of the DDAS: the knowledge base; the reasoning module; and the communication module.



Figure 1. DDAS architecture

Implementation was carried out using CLIPS, an expert system environment developed by NASA and HARDY, a hypertext based diagram editor for Xwindows and windows 3.1. developed by AIAI of the University of Edinburgh.[9,10]

2.2 Knowledge base

The design of the content of the knowledge base, in terms of its organisation and its manipulation is the result of a methodology that goes through four phases:

- 1. the extraction of statements that describe the potential of the modelling techniques
- 2. the extraction of subproblems that the above statements (1) refer to
- 3. the specification of the relationships of the above subproblems (2) with the modelling techniques
- 4. the specification of relationships between and among the above subproblems (2)

2.2.1 Extraction of statements that define the modelling techniques

The modelling techniques were examined in turn and defining statements about each were taken from the relevant literature [1,2]. The statements were descriptions of what a technique does in relation to system usability design, as opposed to how to use an approach.

The statements that were extracted were shown to the modellers to check for interpretation and consistency in order to arrive at a final set of working statements. Examples of "what" statements are given below:

CTA (Cognitive Task Analysis) identifies aspects of design that place heavy demands upon the user's cognitive resources (memory, attention span, etc.)

FSM (Formal Systems Methods) provides a framework for representing and understanding the compatibility between functional (system) state and perceived state.

2.2.2 extraction of subproblems that the above statements (2.2.1) refer to

Considering that for each statement of the modelling techniques, resulting from the previous phase, there is a purpose that justifies its existence, that purpose is used to extract the design subproblem that lies behind the claims of the modelling technique. That is to say, searching for the answers to the question `What problems does each modelling technique try to solve?'.

The CTA example given above implies that there exist aspects of design which are difficult for users to cope with cognitively. In metamorphosing from claims to subproblems, whole sets of subproblems are revealed. It is possible that more than

one statement of a modelling technique refers to the same subproblem. It is also possible that more than one statement from different modelling techniques refers to the same subproblem. In the DDAS, the subproblems are considered in the sense of Checkland's [3,4] purposeful activity subsystems.

These design subproblems are noted down and then compared and contrasted to see which are common in order to arrive to a sets of subproblems which eliminates redundancy.

Some examples of the subproblems are the following:

- isolate features that the user will find hardest to learn
- reason whether some program correctly implements a given specification
- predict reasonable user behaviours that the designer did not intend and did not want etc.

2.2.3 specification of the relationships of the above subproblems (2.2.2) with the modelling techniques.

Each one of the subproblems is related with one or more modelling techniques. Only one type of relationship is considered here, that which specifies how well the modelling technique satisfies the particular subproblem.

For example, a subproblem A may be well satisfied by the CTA modelling technique, somewhat less satisfied by the FSM modelling technique and not satisfied at all by other modelling techniques. This means that the particular subproblem has this `degree of satisfaction' relationship with two of the modelling techniques.

The knowledge about this relationship between the techniques and the subproblems were elicited from the modellers, who were presented with the sets of subproblems and were asked to give a degree of satisfaction of their modelling technique to each one of the subproblems within them. The modellers were given the opportunity to use either a given scale of quantifiers (a lot ... a little) or to define their own scale of quantifiers, in order to give the degree of how well their modelling technique satisfies each one of the subproblems. In the second case, when the modellers defined their own scale of quantifiers, they had to explicitly state what the scale meant. The modellers were invited to make comments on the subproblems, and especially useful were comments regarding the presentation of the subproblems e.g. as disjointed statements, placed in a hierarchy, etc. Examples were taken from case studies to illustrate the subproblem descriptions or modelling technique claims

In the spirit of Checkland, the methodology is viewed as a learning process rather than a requirements/specification process. The modellers were presented with "problems" that were often little more than paraphrases of statements about modelling techniques. The modellers, in rating the subproblems, were asked to ignore these strong associations and to try to assess each subproblem statement as though it were free of connotations. The modellers commented upon the fact that this exercise made them aware of the scope of their techniques as well as how they might be viewed by designers.

2.2.4 specification of relationships among the above subproblems (2.2.2)

Apart from the degree-of-satisfaction type of relationship that each subproblem may have with one or more modelling techniques, the subproblems may also be related with one or more other subproblems. These relationships among the subproblems exist regardless of the modelling techniques. Some of these relationship types are generality/specificity, low possible concurrency, high possible concurrency, and are defined below.

Type generality (gen): if A and B are subproblems, members of the relationship 'A (gen) B', this means that subproblems B is a more specific subproblem than subproblem A and subproblem A is more general than B. In other words, B is a subproblem of subproblem A. This type of relationship includes the type specificity (spec) as well. Each time there is A (gen) B, there is also B (spec) A. Such relationships derive from questions asked such as:

- move towards the specific (How can you tell it's a ...?, Can you give examples of ...?)
- move towards the general (What have ... got in common?, What are ... examples of?, What distinguishes ... from..?)
- move orthogonal to the axis (What alternative examples of ... are there to ...?)

It should be noted that for the following types of relationships A is considered to be one subproblem or a conjunction of subproblems (A1 [and A2 [and ...Ai]]) and B to be one subproblem or a conjunction of subproblems (B1 [and B2 [and ... Bj]]).

Type low possible concurrency (lpc): if A and B are subproblems and also members of the relationship `A (lpc) B', then the possibility that both A and B are parts of the user's problem is very low.

Type high possible concurrency (hpc): if A and B are subproblems and also members of the relationship `A (hpc) B', then the possibility that both A and B are parts of the user's problem is very high.

2.2.5 Results of the methodology

Phase 2.2.3 provides a set of subproblems related to the modelling techniques. Phase 2.2.4 takes this set of subproblems and specifies the relationships among the subsystems-problems that follow the above definitions and thus converts the set of subproblems to a network of related purposeful activity subsystems.

2.3 Interacting with the system

A main assumption of the DDAS is that the designer-user will express his problem according to the subproblems of this network that have their source in the modelling techniques. The following sections discuss the presentation of the subproblems to the user; how the user can be guided to express/identify his problem and what facilities are available for this; and finally how the system outputs recommendations.

The interaction with the user is based upon two types of presentation elements: the graphic display of the subproblems, and the commands that manipulate the interaction.

The subproblems are displayed in the form of labelled shapes and are laid out in a series of screens browsable by the user. Two types of shapes are used, one to represent the fact that there exist more specific problem descriptions in the knowledge base, while the other shape represents the most specific expression of a subproblem contained in the knowledge base. In the current version, the former are shown as rhombii, the latter as circles. Shapes may be linked by arcs which denote different types of relationships existing between subproblems, for example, green arcs represent "high possible concurrency" and blue arcs, "low possible concurrency".

Commands are displayed as buttons on a toolbar which is permanently on screen. These commands aid the user to choose amongst the available facilities of the system, for example the facility of moving to diagrams/screens that correspond to different levels of analysis is performed by double arrow buttons.

2.3.1 Guiding the user to identify his problem

During the interaction the subproblems are displayed to the user, in order for him to search for and identify the subproblems descriptions that he considers as most relevant to his problem. The objective is for him to make a selection of these relevant subproblems which is a way of expressing his situation of concern. Whilst selecting (by left-clicking on the subproblems), the user can also specify the degree of relevance of the subproblems to his problem, and should he change his mind, he can unselect any subproblems he has already chosen. Each time he clicks on a subproblem, its colour changes. Each colour shows the degree of importance of the specific subproblems to the user The set of used colours are white, turquoise, yellow, magenta and red, signifying least to maximum importance respectively. This is also the sequence of the colours which appear when left clicking. After red, (most relevant) comes white again and the user can go through this cycle as many times as he wants.

In order to guide the user through the network of subproblems, these are presented to him at various levels of detailed description. This is done by presenting them in several screens according to the subproblems' degree of generality. It is possible for the user to go backwards and forwards between screens (by using the buttons << >>).

Another feature of the system is that the user can ask for comments from the system about the set of the subproblems he has chosen (Comments on Choices) This facility is available any time during the interaction when selections are made. The comments that the system is able to give are based on the relationships of the subproblems that exist in the knowledge base. For instance, the user who has chosen both of the subproblems that are parts of a "low-possible-concurrency" relationship, is warned that these subproblems are not usually concurrent. The subproblems that are mentioned in the warning messages are highlighted with a black outline in order to find them more easily. The system is flexible in the sense that it allows the user to ignore the warning messages. Should the user want to follow the advice given, he may decide how he wants to solve the implications, by either selecting and unselecting accordingly. Once all the warning messages that the system has to show according to the relationships have been displayed, the system reverts to the normal interaction state where the user can choose/unchoose subproblems or choose one of the other available facilities.

A further facility available at any time is that of providing a formatted text description of the set of subproblems chosen (Current State). The relationships that exist in the knowledge base form the basis for the text description of the chosen subproblems. Once the text description of the set of chosen subproblems has been presented, the system reverts to state where the user can choose/unchoose subproblems or choose one of the other available facilities.

Should the user want an illustration of a particular problem description, he can obtain examples of use of the he by using the example button (or by shift-left-clicking on the subproblems in question). This feature can be useful in helping the user decide about how close (if at all) the specific subproblem description is to his own particular problem.

Finally, the user is also able to see instructions regarding the use of the system (Help button).

2.3.2 Output of the interaction: Recommending

When the designer-user feels that the subproblems he has chosen describe his problem situation adequately, he can request a recommendation from the system.

The DDAS recommends which the modelling technique(s) are suitable for his problem. This facility is available whenever no other facility is active.

In the current version of DDAS, the recommendation is a formatted text which recommends the user the most appropriate technique(s). The reasoning behind this recommendation, which is based upon fuzzy logic, is also given in the formatted text, in order to give the user the justification of the rationale behind the recommendation. The compensation oriented score operator from test score semantics [12] is used to compute the recommendation. For its computation the degrees that specify how important each chosen most specific subproblems is to the user (3.2.) and the degrees of how well the modelling techniques satisfy each chosen subproblems are used (2.3).

3. Example

To illustrate some of the system's capabilities, an example detailing a designer's specific problem and how he can handle it using DDAS follows. In this example, a designer's concern is that interface users are often confused by the outcome of clicking a button X. e.g. there can be two different results of clicking the same button X in two different contexts respectively.



Figure 2. A snapshot of the most general subproblems diagram

The designer wants to resolve this problem. For the sake of the example, it is assumed that he wants the solution to enable the users of the interface to distinguish clearly what are the corresponding effects on the system when a button is pressed. It is also assumed that the designer wants to check that this problem of the design of the interface does not start from a confusion in the requirements.

The designer is firstly presented with a diagram which uses rhombii to represent the most general subproblem descriptions, such as that given in Fig 2. The designer searches through the diagram for labels which come closest to expressing his problem. In this case, he chooses the rhombii with the following labels and assigns to them a degree of relevance:

- IDENTIFY FEATURES IN THE DESIGN OF THE INTERFACE THAT NEED MODIFICATIONS OR EXTENSIONS (red)
- IDENTIFY PROBLEMATIC FEATURES IN THE REQUIREMENTS (yellow)
- PROVIDE A FRAMEWORK FOR CAPTURING PROPERTIES THAT ARE GENERALLY REQUIRED TO EXIST BETWEEN THE SYSTEM AND THE INTERFACE (magenta)



Figure 3. A snapshot of the most specific subproblems diagram

The designer presses the button with the label «>>« in order to move to the next diagram with the more specific subproblem descriptions. He is then presented with a diagram which uses circles and arcs to represent the possible subproblem descriptions and the relationships between them, such as that given in Fig 3. The designer searches through the network diagrams for labels which come closest to expressing his problem. In this case, he chooses the circles with the following labels and assigns to them a degree of relevance (colour).

- identify features that are sources of ambiguity and confusion (red)
- identify ambiguities and confusions in the requirements and therefore iterate towards design specifications that are cognitively straightforward (yellow)
- provide a framework for representing and understanding the compatibility between functional (system) state and perceived state (conformance) (magenta)
- provide a framework for representing and understanding the trade-off between what the representation in itself will support and what must be supported by the system (affordance) (turquoise)

• provide a framework for representing and understanding the property of predictability: supporting the system tasks by providing enough information to indicate to user what effect his new actions will have (magenta)

Before going on to choose some more subproblem descriptions from the DDAS diagram, the designer would like to have a commentary from the system about his choices. He clicks on the «COMMENTS ON CHOICES» grey button. This advice is given in a message window as shown in Fig. 4.



Figure 4. Comments on Choices message window

In this particular case the displayed message comments that according to the system, the subproblem description *«provide a framework for representing and understanding the compatibility between functional (system) state and perceived state (conformance)»* usually implies the one with the label *«provide a framework for representing and understanding the feedback which shows that a mistake has been made and the ease with which an inverse for an incorrect action can be found (repair and recovery)»* and therefore the second could also be chosen.



Figure 5. Message window with the available example(s) for the chosen specific subproblem

The designer can shift-left-click on a subproblem in order to see the available examples (if any) of the specific subproblem. The examples help him understand some characteristic situations that the subproblem should be chosen. The examples are given in a message window as shown in Fig. 5.

Each time the designer wants to see a text description of the chosen subproblem he clicks on the «CURRENT STATE» grey button. A window appears with formatted text which consists of sentences that contain either one selected subproblems description or two selected subproblems that are related with a type of relationship expressed in words. In this example, a part of the text description that the designer sees is shown on the window in Figure 6.

In this way, the system, utilising its knowledge of the design space, and subproblems associated with it, prompts the user and aids him to consider subproblem descriptions which may be relevant to his problem of concern and which he has not chosen. The user considers the system's advice and is free to reject it should he not think it relevant.



Figure 6. Current state message window

Otherwise, the system highlights the subproblems mentioned with a black outline (Figure 7) to help the user find the subproblems that the message refers to.

The user continues in this way, making selections, reading the comments on current choices and reselecting until he is satisfies with what the current selection represents. During this cycle he can get at any time a text description of the current state.

When the designer is satisfied that he has a final set of chosen subproblem descriptions (i.e. he doesn't want to choose any more subproblem descriptions by clicking on them and that he doesn't want to change his belief about the importance he gave to the selected subproblem descriptions, by changing their colour), he then clicks on the «Recommendation» button to get a recommendation about the most appropriate modelling technique(s) for his problem. A window appears with the recommendation. The computation representing the reasoning behind this result is also displayed in the same window for traceablity. This can be transformed to formatted text, in order to give the user the opportunity to understand and justify the system's reasoning (Figure 8).



Figure 7. Selected subproblems



Figure 8. Recommendation message window

4. Conclusions and Discussion

Making a generally applicable catalogue of design problems, and/or even making a taxonomy of such problems, is a difficult task to undertake and it is doubtful, in view of the rapidly changing nature of technology, and users' responses to it, whether such a task could be achieved satisfactorily. The DDAS concentrates on those design space subproblems which can be aided by the modelling techniques and upon classifying these into a network according to the relationships that exist among them.

This approach does not compare and contrast the modelling techniques. Instead, the methodology looks at what design problems these techniques are capable of dealing with and works by comparing and contrasting the problems, not the techniques themselves. In this way a network of subproblems is generated. The network and the relationships within it relate to the degree with which a modelling technique can deal with a subproblem, as well as to the relationships between the subproblems themselves. When a user selects a sample set which most closely resemble his own situation of concern, the system reasons by means of the test score semantics using a number of operators, to provide a recommendation as to which modelling technique(s) are the most appropriate for the user's particular problem, and backs this up with a justification of its rationale.

The importance of the DDAS, lies not just in its usefulness to the designer, who now has access to bodies of knowledge in direct relation to a problem of concern, but in its claim to provide a methodology for decision aid in similar situations where problems exist, and tools to solve them also, but where a short cut or aid is needed to bring the two together.

5. References

- [1] Amodeus, ESPRIT Basic Research Action 3066, 1989-1992, AMODEUS (Assimilating Models of Designers Users and Systems) and ESPRIT Basic Research Action 7040 AMODEUS II 1992-1995 (Assaying Means of Design Expressions for Users and Systems) Documentation available by anonymous ftp (ftp.mrc-apu.cam.ac.uk) or by www (http://www.mrcapu.cam.ac.uk/amodeus/gref.html).
- [2] Buckingham Shum S., Jørgensen A.H., Hammond N. and Aboulafia A.(Eds), Amodeus-2 HCI Modelling and Design approaches: Executive Summaries and Worked Examples, *Amodeus Project Document: TA/WP16.*, 1994
- [3] Checkland P.B. Systems Thinking, Systems Practice, Wiley, New York, 1981.

- [4] Checkland P.B., Scholes J. Soft Systems Methodology in action, Wiley, New York, 1990.
- [5] Darzentas J., Darzentas J.S., Spyrou T. Defining the Design "Decision Space": Rich Pictures and Relevant Subsystems, Amodeus Project Document: TA/WP 21, 1994.
- [6] Darzentas J., Darzentas J.S., Spyrou T. Fuzzy Reasoning and Systems Thinking in a Decision Aid for Designers in Proceedings of Second European Conference on Intelligent Techniques and Soft Computing, Aachen, pp1609-1614, 1994
- [7] Darzentas J., Darzentas J.S., Spyrou T. Designing a Designers' Decision Aiding System (DDAS): a Designers' Decision Aiding System, Journal of Decision Systems Hermes (in press)
- [8] Darzentas J., Darzentas J.S., Spyrou T. An Architecture for Designer Decision Aiding in Brannback, M. and.Leino, T (Eds) DSS-Galore! Ebo Academy Press, Ebo Ser. A. 427, pp115-132, 1995.
- [9] Giarratano J. and Riley G. Expert Systems: Principles and Programming. PWS Publishing, Boston, MA., 2nd. edition, 1994.
- [10] NASA Johnson Space Center, Houston, TX, "Clips Programmer's Guide, Version 6.0, JSC-25012", June 1993.
- [11] Raghav Rao H., Sridhar R., Narain S. An active intelligent decision support system - Architecture and Simulation. *Decision Support Systems*, 12, pp. 79-91, 1994
- [12] Zadeh L.A. Knowledge Representation in Fuzzy Logic, *IEEE Transactions on Knowledge and Data Engng* 1 n° 1, pp. 89-100, 1989.

Combining Techniques from Intelligent and Decision Support Systems: An Application in Network Security

Thomas Spyrou¹ & Rainer Telesko²

¹University of the Aegean, Department of Mathematics Research Laboratory of Samos GR 83200 Karlovassi, Samos, Greece

²University of Vienna, Institute for Applied Computer Science Department of Knowledge Engineering AT 1210 Vienna, Austria

This paper presents the design and development of a prototype of an expert system application for the detection of certain types of abnormal behaviour in open networks (and in particular in the management systems of such networks) and shows how it provides decision aid for the System Security Officer (SSO). The objective is to aid the SSO determine the seriousness of the possible attack and also to help him to find and apply the appropriate countermeasures. The case of unknown attacks is discussed here, where the SSO has to consider a number of constraints. Sets of actions acceptable by the system but which at a higher level may form a non acceptable task or tasks are assumed to be detected by the User Intention Identification (UII) module as potentially malicious intentions. The Decision Module (DM) is, responsible for aiding SSO with his decision regarding possible intrusion detected by the UII module. These two modules are presented and discussed and the way these two modules communicate is also introduced. This work is carried out under SECURENET, a project that aims at the protection of networks and in particular their management. The development of the current version of the modules was carried out in the frame-based expert system-shell CLIPS.

Acknowledgement: The work reported on in this paper was funded by the RACE - SECURENET II (R2113) project.

1. Introduction

This paper presents the design and development of a prototype of an expert system application for the detection of certain types of abnormal behaviour in open networks (and in particular in the management systems of such networks) and shows how it provides decision aid for the System Security Officer (SSO). This work is being carried out under SECURENET [7,8], a project that aims at the protection of networks and in particular their management. Such systems are open to various types of malicious attacks and intrusions which in most cases consist of a few or many illegal acts, and their detection usually triggers appropriate countermeasures. The present work is concerned with detection of that type of attacks/intrusions which do not usually consist of illegal actions, but of a set of actions acceptable by the system, but which at a higher level may form a non acceptable task or tasks. This form of intrusion is regarded as part of users' intentions about the use of the system, i.e. the tasks they intend to perform.

The objective is to aid the SSO determine the seriousness of the possible attack and also to help him to look for the appropriate countermeasures. This is fairly straightforward when a known attack is the issue but in the case of unknown attacks the SSO has to consider a number of constraints.

Two modules of the system are presented and discussed, the User Intention Identification (UII) module [9] and the Decision Module (DM) [3] and the way these two modules communicate is described.

The development of the current version of the modules was carried out in the frame-based expert system-shell CLIPS [2,4].

Section 2 gives an overview of the SECURENET system and its architecture. In section 3 and 4, surveys of the User Intention Identification module and the Decision module are given. Section 5 discusses the way these two modules are combined to produce a decision aid for the Security Officer of a network system and finally a section with conclusions and discussion follows.

2. An overview of SECURENET.

The SECURENET-project is developing a network-monitoring and analysissystem which aims at networks protection. It is composed of several modules with different responsibilities. According to the SECURENET II-architecture there are three necessary steps a system must perform in order to detect malicious attacks in a network: the first step where the network activity is monitored, the second step where this monitored activity is analysed according to a number of techniques, and the third stage where the results of this analysis are evaluated and the seriousness of the danger is estimated. If there is much evidence that there is an (ongoing) intrusion, appropriate countermeasures will be selected in the Countermeasure module - and when there is the 'O.K.' from the SSO - executed.

Figure 1 presents the SECURENET II architecture with the main modules and the respective interfaces.

The heart of the SECURENET-system according to this architecture is the analysis module which gathers information coming from the actual network in order to recognise or to infer malicious attacks to the network. Currently there are three analysis modules, the Neural Network module, the Expert System module and the User Intention Identification module. A Decision support module is also necessary in order to further elaborate the results of the analysis made by the analysis modules.





Finally there are several monitoring modules which observe the real network acquiring the information necessary for the analysis step. All these modules need to

pass information each other in an efficient way. An intermodule communication system (SICS) [10,11,12] is used to play the role of communication manager and in parallel it has the responsibility to maintain a common communication dictionary.

3. Overview of the User Intention Identification Module (UII)

The User Intention Identification module is an autonomous module for the detection of anomalous behaviour by reasoning about the characterisation of the intentions of users. This module views the users of a system as using it in order to achieve certain goals by performing various tasks. This module plays a complementary role within the SECURENET-system, trying to detect a range of malicious attacks with special characteristics. The major characteristics of the malicious attacks that this module aims at, are those cases where malicious tasks are composed of legal events. Since the basic actions for these tasks are allowable they cannot be detected by simple matching mechanisms. The examination of the whole rationality behind the execution of these basic actions has to be considered in relation to the general goals these actions are trying to fulfil (when composed to form tasks). Reasoning about the deviations observed in the execution of actions within a task in relation to the normal task execution (under the general goaloriented constraints) offers an indication of the suspiciousness of the observed behaviour. This reasoning mechanism is based on a semantic matching of observed user activity with a representation of normal behaviour. Task related Knowledge structures (TKS) is utilised here for the representation of the normal behaviour of the users of a system when they perform a number of tasks.

In other words the UII system aims to detect a certain type of malicious attacks by characterising the normality of the behaviour of the users. Based on the knowledge representations described above the system represents the expected normal behaviour and compares it against the current, observed behaviour. Deviations correspond to abnormality and a certainty factor is calculated as an indication of the importance of this abnormality.

As a system, UII receives as input the action that the user is executing, the time of execution and the user identity and gives as output an indication of the suspiciousness of the observed behaviour and an evaluation of executed tasks.

Figure 2 gives an outline of the architecture of the User Intention Identification Module and its components which are discussed in the sequel.



Figure 2. Architecture of the UII module

3.1 Task and Functionalities

The main objective of the User Intention Identification Module is the semantic interpretation of the collected data that correspond to the network users' actions. According to this the Intention Identification module must be able to perform four major tasks.

- Pre-processing of the input data
- Task Synthesis.
- Behaviour Comparison.
- Output

There are four functions corresponding to these four tasks:

3.1.1 Audit Data Processor (ADP)

The input data for the UII is received/provided by the audit file. This audit file remains open during the process of identification of the tasks for execution (or for possible execution) by the UII. This file receives information about the user identity, the actions are executed in the net; this process is continuity. Every time that an identification has finished, a new audit is ready to be read by the UII. Sets of rules are triggered to determine the task or tasks to which this observed action can be attached.

3.1.2 Task Synthesiser

This function implements the first steps of the semantic interpretation of the behaviour of the observed network users. This interpretation is the basic function of the Intention Identification module. TKS entities identified by the ADP function are characterised as task components. Sets of rules are triggered to determine the task(s) to which these TKS entities can be attached. These rules represent the relations of the observed actions within and between the possible tasks executed by the observed entity. These relations are represented as pre-conditions or post-conditions, sequence relationships, association to wider tasks and goals etc.

The basic idea behind this function is that a first comparison has to be made in order to discard non valid hypotheses about the execution of tasks. This is necessary in order to let the comparator function work with valid tasks only. The examination of the TKS entities in this function is made primarily for the validation of an action as a part of a task.

3.1.3 Comparator

This is the main inference mechanism of the module. It makes all the semantic interpretation of the observed network user behaviour and produces the intrusion hypotheses. The basic idea behind this function is that by reasoning about deviations from the normal task execution and by reasoning about the similarities of the executed with allowable tasks, estimations regarding the suspiciousness of the performed activity could be made. These hypotheses are formed in the output function.

The reasoning mechanism of this function is realised with sets of rules that represent the relationships between the various parts of a TKS. The TKS knowledge structure is used as the knowledge base for that function and knowledge about task execution is represented with that knowledge structure. There are three major aims for that function:

- It tries to combine the relations between the TKS entities in order to decide about the normality and validity in a Task execution
- It tries to combine inter-task relations and associations in relation to relevant

T. Spyrou and R. Telesko

goals in order to decide whether a combination of task execution is suspicious and finally

• It tries to combine the goal substructure of the various TKS structures of tasks executed with within and between role relations in the tasks execution. This is the most complex aim of the comparator function and for the purposes of the demonstrator the basic role relations will be examined.

The basic operation of this function is performed by a recursive process which elaborates the TKS structure representation in relation to the asserted facts that correspond to observed behaviour and come from the previous two functions.

The output of this function is a hypothesis that classifies a suspicious situation observed and provides the data for a representative description of the reasoning that produced that hypothesis.

3.1.4 Output

This function plays the role of the explanation generation part of an expert system. Every time that a hypothesis is generated by the comparator function this function selects the necessary information relevant to this suspected intrusion from the TKS based profile and offers it as a means of explanation. This information is passed to the Decision Module.

4. Overview of the Decision Module (DM)

The main task of the DM [3] is to produce a final decision about whether an attack has taken place or not. Figure 3 presents an outline of its internal architecture.



Figure 3. The architecture of Decision Module

The input to the DM are various hypotheses and confidence values delivered from the

- Expert System module (ES)
- Neural Net module (NN) and
- User Intention Identification module (UII).

4.1 Input Data Processor

The DM is triggered if at least one of the three detection modules (ES, NN or UII) sends a messages to the DM. The messages from the three detection modules are collected in a queue according to the FIFO-principle and have all the same general format (Table 1).

Field	Format	Description
system_id	string[8]	Operating system identifier
node_id	long	node identifier number
time	long	record producing time/date
record_id	long	serial number for record
user_id	long	unix audit user identifier
hypothesis_type	long	type of hypothesis
level_of_conf	long	level of confidence
param_number	long	number of additional parameters
params	char array	parameters

Table 1: General format of ES/NN/UII-messages
Most important for the DM are the fields *hypothesis_type*, which contains the generated hypothesis, *params*, which gives a list of the 'statements' that led to the alarm in the respective module and *level_of_conf*, which associates a confidence value between [0,...,9] for that hypothesis. '0' means no confidence, while a value of '9' shows high confidence in the generated hypothesis. The task of the DM is now to *confirm* or *reject* the previous generated hypothesis. In the first case the *level_of_conf* will be increased, in the latter case decreased. If there is no additional information available in the DM, the *level_of_conf* will remain unchanged. Two methods are available for performing this task:

4.2 Analysing the Statement list:

First of all one has to verify that the alarm raised by a module is not a 'false alarm'.

A one possible way to do this is simply analysing the statement-list (user commands). For example, if some critical statements (e.g. open file, write file, chmod, rm) or a critical 'combination' of statements is found, the level_of_conf will remain unchanged if not raised.

4.3 Hypothesis comparator:

If two or more alarms are raised in the same time period (this can be checked by 'temporal reasoning'), the highest level_of_conf will be delivered to the SSO and the Countermeasure module in order to make sure that the problem is not underestimated.

The field attack classes will be determined by matching the facts with previously defined rules. These rules are related to the targets of an intrusion and are domain-dependent. A small example is shown in section 5 ('reroute example').

4.4 Output message processor:

The output of the Decision Module is coded according to the DMMSG format (Table 2). This output is communicated with the Countermeasure Module and it is also passed to the System Security Officer.

Field	Format	Description
system_id	string[8]	Operating System-identification

node_id	long	node identifier number
time	long	record producing time/date
record_id	long	serial number for record
user_id	long	user identifier
level_of_conf	long	level of confidence
level_penet	int	level of penetration
attack_classes	long	attack classification
param_number	long	number of additional parameters
params	char array	parameters

Table 2: Format of DMMSG

The most important fields for the SSO here are the previous mentioned level_of_conf and the attack classes. The latter field (Table. 3) contains information (on a high level), about which sort of intrusion the SSO and the Countermeasure Module have to face with.

Attack class	Symptoms
No Attack	Used only by the DM for the report and log file.
Trojan Horse	Unexpected file operations, inappropriate source code,
	unexpected communications
Logic Bomb	Inappropriate source code, operations on sensitive
	resources
Insider attack	user operates outside user model thresholds, operations on
	sensitive resources, inappropriate creation and
	manipulation of data instances
Password Cracking	Repeated failed attempts to establish another identity,
	operations on passwd-file
System	attempt to exploit known weaknesses in a system routine
programming	
attacks	
Outsider access	Repeated failed access to establish an identity; attempts to
violation	use default system passwords
Denial of service	Unexplained loss of contact with victim system, large
	amounts of meaningless traffic jamming available
	bandwidth
Trapdoor attack	inappropriate source code, operations on sensitive
	resources
Known attack	Use of known system bugs and -loopholes

Table 3: Attack classes in SECURENET

132

5. Producing useful aid.

This section describes the way the two modules UII and DM communicate information in order to produce a useful decision for the security officer. In other words the collaboration of a module that analyses observed behaviour and a module that utilises this information for use in the real world.

The whole functionality of the communication will be shown through an example where a produced hypothesis by the UII module triggers the DM.

5.1 Checking the UII-status:

First of all, the current state of the UII module has to be examined in order to estimate the weight of the results of the analysis.

A fuzzy variable called uii_state to hold the current status of UII is introduced:

uii_state = {UNSTABLE, WEAK, NORMAL}

uii_state has to be set to UNSTABLE if UII is out of operation (e.g. the knowledge base is updated with new tasks). uii_state will be WEAK if UII has produced an unusual number of false alarms in the past. If UII works quite normally, then uii_state gets a NORMAL.

The value of uii_state is stored in the DM configuration file, which will be read anytime the DM is started and updated and anytime the DM terminates. Furthermore the SSO will be informed of the actual entry of uii_state. For example if there is a WEAK and there comes an alarm from the other analysis modules related to the same time period, this has the practical consequence that UII has not the same 'weight' in the subsequent decision process. The more false alarms are produced the less will be the weight of this module.

5.2 Analysing UII-hypothesis:

Let us assume that uii_state has a NORMAL and the following examplehypothesis is delivered within an UII-message to DM:

Hypothesis	Description of Hypothesis	Parameters
600400	The hypothesis is produced when there is	Possible executed
	strong evidence from the history of the user	tasks, conflicted
	that during the execution of a number of tasks	strategies/goals
	there exist crucial strategy or goal conflicts.	
	These conflicts are identified after the	
	comparison of the current history of the user	
	with the knowledge in the knowledge base	
	about the tasks/goals relations	

Example hypothesis:

(Only relevant information is given below)

Hypothesis-number (rule set) : 600400 level_of_conf: 5 task: remove a routing node for service strategy: 1) check system state 2) reroute traffic 3) remove routing node 4) check system state params: reroute traffic # params: 1

5.3 Composing the DMMSG:

According to the analysis done in the UII module one can see that the system administrator doesn't follow the normal way of removing a routing node from a network. It may now be possible that there exists a so-called 'insider attack', that is, if someone operates outside user model thresholds, operations on sensitive resources, inappropriate creation and manipulation of resources.

This assumption is coded in the following DM-rule (written in pseudo-code):

```
IF (hypothesis-number = 600400)
THEN
    attack classes = 'insider attack'
    level_of_conf = unchanged
    level_penet = '2'
    params = [service; insider]
```

The level_of_conf remains unchanged because there is no additional knowledge (in the DM) available that there is really an 'insider attack'. It may well be that the administrator acts in a very urgent situation and it is not possible to follow the normal procedures

The level_penet which is an indication for the gravity of an intrusion is set to '2' (Table 4). The field params contains additional information for the SSO; 'service' means that a service (routing) is the target of the actual intrusion, 'insider' specifies the intruder which has in the actual case, a valid account for the network.

Level 0	INFORMATION, NOTIFICATION
Level 1	ATTEMPT of an intrusion which does not succeed
Level 2	ABNORMAL BEHAVIOUR of a user but no more information
Level 3	DAMAGE LIMITED to non-privileged users
Level 4	Intruder gets PRIVILEGED RIGHTS

Table. 4.: Instances of	of level_penet
-------------------------	----------------

To make sure that there is no misuse of hardware resources the SSO and the Countermeasure Module will be informed. The SSO can talk to the user (which has been identified via the UII-record) and can find out what's going on.

6. Summary and discussion

In this paper the design and development of a prototype of an expert system application for the detection of certain types of abnormal behaviour in open networks has been presented. Two modules of the system have been presented, the User Intention Identification (UII) module and the Decision Module (DM).

The User Intention Identification module is a module for the detection of anomalous behaviour by reasoning about the characterisation of the intentions of users. This module views the users of a system as using it in order to achieve certain goals by performing various tasks. This module aims to detect a range of malicious attacks i.e. in the cases where malicious tasks are composed of legal events. Since the basic actions for these tasks are allowable they cannot be detected by simple matching mechanisms. The examination of the whole rationality behind the execution of these basic actions has to be considered in relation to the general goals these actions are trying to fulfil (when composed to form tasks). Reasoning about the deviations observed in the execution of actions within a task in relation to the normal task execution (under the general goal-oriented constraints) offers an indication of the suspiciousness of the observed behaviour

The main task of the Decision Module on the other hand, is to produce a final decision about whether an attack has taken place or not. According to its

architecture this module receives input from various analysis modules of the system and tries to combine this input in order to provide useful aid to the SSO. The objective is to aid the SSO determine the seriousness of the possible attack and also to help him to look for the appropriate countermeasures. This is fairly straightforward when a known attack is the issue but in the case of unknown attacks the SSO has to consider a number of constraints.

One of the main problems during the design and development of the system was the communication problems between the two modules. These communication problems range from the actual implementation of the data exchange between the two modules to the determination of the limits where the behaviour analysis finishes and the decision making process starts.

The development of the current version of the modules in the frame-based expert system-shell CLIPS was proved to be very successful. The speed of the running system is fairly adequate and an evaluation process has been planned and is now being carried out.

7. References

- J. Darzentas and T. Spyrou. Functional Specifications of SECURENET Components, In: SECURENET System Development Plan, CEC RACE Report, R2057.EXP.DR.L.050B1, pp. 53 - 143, 1992.
- [2] J. Giarratano and G. Riley. Expert Systems: Principles and Programming. PWS Publishing, Boston, MA., 2nd. edition, 1994.
- [3] D. Karagiannis, C. Mayr, R. Telesko: Design of the Decision Module, Deliverable Nr. 17, Internal Publication of the RACE-Project R2113, February 1995.
- [4] NASA Johnson Space Center, Houston, TX, "Clips Programmer's Guide, Version 6.0, JSC-25012", June 1993.
- [5] SECURENET Project, Deliverable 3, Overall System Concept, Technical Report R2057.EXP.DR.L.030B1, RACE Programme, 1992
- [6] SECURENET Project, Deliverable 5, System Development Plan, Technical Report R2057.EXP.DR.L.050B1, RACE Programme, 1992
- [7] Spirakis P., Katsikas S., Gritzalis D., Allegre F., Darzentas J., Gigante C., Karagisnnis D., Kess P., Putkonen H. and Spyrou T., SECURENET: A Network Oriented Intrusion Prevention and Detection Intelligent System, *Network Security Journal*, vol. 1, no 1, Nov. 1994.
- [8] Spirakis P., Katsikas S., Gritzalis D., Allegre F., Darzentas J., Gigante C., Karagisnnis D., Kess P., Putkonen H. and Spyrou T., SECURENET: A

Network Oriented Intrusion Prevention and Detection Intelligent System, IFIP SEC93, *Proceedings of the 10th International Conference on Information Security*, May 1994.

- [9] T. Spyrou, J. Darzentas. Specifications and Design of the Intention Identification Module, Deliverable Nr. 17, Internal Publication of the RACE-Project R2113, December 1994.
- [10] E. Sutinen, "/DEL/IMP/4.2/OULU/EPS/050395; Implementation of SICS", *Securenet II deliverable*, 1995.
- [11] E. Sutinen and H. Putkonen, "/DEL/SPC/4.1/OULU/EPS/050594; Specification of SICS", *Securenet II deliverable*, 1994.
- [12] E. Sutinen and H. Putkonen, "/TRP/DES/4.2/OULU/EPS/050395; Design of SICS", Securenet II deliverable, 1995.

More Effective Strategic Management with Hyperknowledge: Case Woodstrat

Pirkko Walden and Christer Carlsson

Institute for Advanced Management Systems Research (IAMSR) Åbo Akademi University DataCity A 3208, 20520 Åbo, Finland

Woodstrat is a support system developed for strategic management. The development process was carried out interactively with managers in two major Finnish forest industry corporations. The system is modular and is built around the actual business logic guiding and controlling strategic management in these two corporations. The main modules cover the market position, the competitive position, the productivity position and the profitability and financing positions. The innovation in Woodstrat is that these modules are linked together in a hyperknowledge fashion, that is, the core concepts of strategic management are brought to interact. The intermodular links are based on expert knowledge which also is worked into the modules to guide the manager through the process of working out sustainable competitive advantages. The hyperknowledge approach makes this support intuitive and effective, as the elements are linked to each other in combinations that are familiar to the user.

Key words: Strategic management, hyperknowledge, decision support systems, interactive support, forest industry

Introduction

The process of *strategic management* is about coping with complex relationships and uncertain futures in a way that enables us to create and maintain sustainable competitive advantages for a corporation and its strategic business units. This process is a dynamic one and sensitive to changes in the competitive context, at least that is what it should be. In practice, however, it has turned out that the process is often a formalistic exercise which offers very little substance to its participants. One of the many reasons for that is that rapidly changing markets and a strong competition quickly make most strategic plans obsolete; thus most managers do not think it worth too much effort to build comprehensive plans. There are, however, some research results which show that a lot of companies can be expected to improve their strategic management processes with the help of knowledge-based support systems (1). There are a number of moments which could be handled more effectively and with much better results if some support systems technology would be applied.

Among managers *strategic management* is normally understood to cover both the strategic planning process and the implementation of its results. It is an integrated program of means by which a firm secures and sustains competitive advantages (2). This involves understanding and transforming some proper and consistent selection of strategic management concepts into specific strategic action programs.

In our specific context, the Finnish forestry industry, this translates into working out how to position strategic business units of a corporation such that the corporation can build and sustain competitive advantages in its key market segments.

As a starting point, let us first construct a conceptual skeleton of what could be the substance of strategic management:

strategic management is the process through which a company for a chosen planning period (i) defines its operational context, (ii) outlines and decides upon its strategic goals and long-term objectives, (iii) explores and decides upon its strengths, weaknesses, opportunities and threats, (iv) formulates its sustainable competitive advantages and (v) develops a program of actions, which exploit its competitive advantages and ensure profitability, financial balance, adaptability to sudden changes and a sound development of its capital structure.

It is quite easy to verify that this formulation is consistent with most definitions given by various authors (cf. 2,3,4,5).

We used this fairly rough formulation as a basis and gradually worked out a joint understanding of the elements with the management teams in 15 strategic business units (SBUs) from both corporations. We actually did not have too much debate about the conceptual basis and the key concepts (the only exception was some of the corporate planning staff who were more thorough on the definitions than either we or the SBU managers). The focus was on practical issues such as (i) where to get reliable external data, (ii) how to find enough information about strengths and weaknesses of key competitors, (iii) how to determine the effects of perceived competitive advantages on competitive positions in key market segments, and (iv) how to combine and transform the results of strategic decisions to estimate profitability, financial position and capital structure for the strategic planning period. We gradually learned that the way managers think about strategy, and make sense of their company worlds, is the basis for creating strategic visions. Visions are here seen as embedded in strategic management, both in the creation and the implementation of what we define as sustainable competitive advantages. When developing *Woodstrat* we helped the managers to form visions of how to create and articulate *sustainable competitive advantages;* these were understood as a synthesis of visions of a *market position, competitive position, production position, investments* and their related *financing;* the visions are enhanced or restricted by facts about the context, which identify the market potential, key competitive factors, the competitors and the set of possible strategies (cf. *environment, competitors, product mix*); the visions and the facts are combined to generate options and form strategic action programs, which are defined both for the strategic business unit and corporate levels; finally, the visions and the strategic action programs are evaluated in terms of profitability and capital

structure (specified as an *income estimate*, a *balance estimate*, *long term cash flow* and *key ratios*) which represent stakeholder interests.

The strategy formation process in *Woodstrat* helps a strategy emerge from a qualitative belief system (cf. figure 1).



Figure 1: Strategy Formation Process

The notion of an *emerging strategy* was introduced by Mintzberg (5) and we tried to evaluate this process in a series of studies with the *Woodstrat* in 1993-94: (i) in interactive model building seminars with the managers, in which the basic logic of the system was formed; (ii) in studies of the knowledge bases used for actual strategic planning, and (iii) in workshops with the managers, in which they built their strategic plans.

The strategy formation process shown in figure 1, is quite close to Eden's (6) formulation; he shows a practical way to cope with complexity in an orderly and systematic fashion, allowing even a theoretical basis for the approach (Kelly's personal construct theory) and proceeding to demonstrate that qualitative belief systems can both be analyzed and synthesized; he, furthermore, demonstrates that this can be done with computer support. Working with the *Woodstrat* also we have seen that modern information technology has made it possible for line managers to use formal procedures to form novel strategies for their key products in their most important market segments; this in real time and in anticipation of strategic moves by their key competitors.

The Woodstrat system is a hybrid object-oriented expert and hyperknowledge system which was built to serve as a support system for strategic management. We will show that the hybrid system allows both systematic modeling and adaptive, interactive learning; then we will demonstrate that the system is useful as a support system for strategic management. This paper is a continuation of previously published research work (cf. 7-15).

A Hyperknowledge Environment

The idea of creating a hyperknowledge environment is a fairly recent one (cf. 16-18). The first systematic discussion of hyperknowledge was published by Chang, Holsapple and Whinston (cf. 16) in 1993. They introduced the principle that a decision support system (DSS) should form a "hyperknowledge environment" with its users, that is, the DSS should be an extension of the user's acquired knowledge management capabilities. The decision maker is described through a cognitive metaphor: a decision process is carried out by navigating through a universe of concepts. Some of these concepts are descriptive, some are procedural, and some are context dependent, abstract goal formulating and motivating concepts which serve as instruments to forge a joint value and goal system. The ideal DSS for this purpose is a knowledge-rich environment for the user, in which he can access and manipulate concepts and work out their interdependencies. The DSS is built in such a way that the interdependencies represent the internal logic of the context the user tries to understand and tackle.

Chang et al. (cf 16) have formulated "the fabric of hyperknowledge" in a series of 13 propositions, which are based on the Bonczek-Holsapple-Whinston conceptual

framework (19); as this is rather outdated - it was published in 1981 - and builds on ideas derived from a software technology more than two generations old, we will use the substance, not the technological constructs of the propositions (proposition 13 is omitted as it is inconsistent with the other 12):

- 1. a knowledge system (KS) is built from concepts, each of which can be referenced by a unique identifier; a problem processing system (PPS) should be built in such a way that it can operate on the identifiers;
- 2. there exists a concept map in the KS which shows all the definitional relationships between the concepts;
- 3. there exists a functional map of the concepts which together with (2) helps the PPS navigate in the KS;
- 4. there exists an association function in KS such that each concept can be associated with any give concept; in any association at least one concept is an *agent* and at least one other concept an *object*;
- 5. each association function is differentiated from all other associations in KS;
- 6. for each assosiation function agents and objects can be identified;
- 7. associations can be inherited;
- 8. each association belongs to a specific cardinality class;
- 9. it is possible to build functions for focusing on a particular concept from a set of concepts to satisfy a user's contact needs;
- 10. it is possible to build functions to send messages to any contacted concept; each message can be built to produce a valid kind of impact;
- 11. the result of impacting a concept is an output message;
- 12. for a full-fledged decision support environment the problem solving function must have all the properties 1-11;

This description is, on an intuitive level, quite close to the environment we created in *Woodstrat* (cf. figure 1). There are some useful characteristics of a hyperknowledge environment (cf. 16): (i) the user can navigate through and work with diverse concepts; (ii) concepts can be different epistemologically, (iii) concepts can be organized in cognitive maps, (iv) the concepts can be made interrelated and interdependent, (v) relations can be structured or dynamic, and (vi) relations can change with or adapt to the context. There are also a couple of problems with hyperknowledge (cf. 17): (vii) it has turned out to be too informal and unstructured for handling complex problems, and (viii) users get lost in a conceptually over-rich environment, i.e. they lose touch with the task they try to accomplish. Some ways to handle these problems have been shown in (9), and have been incorporated in the knowledge-based support system discussed here.

Woodstrat

Woodstrat was built around the actual business logic recorded in 15 SBUs of two forest industry corporations. The system was developed as a series of prototypes in 1992-94; after the first versions with a Lisp-based expert systems shell proved to be too inflexible, the next versions were built with Toolbook, which introduced the notion of *hyperknowledge*. The present full-scale system was built as a hybrid system in Visual Basic in which the features of the Lisp and Toolbook versions were rebuilt as objects. With Visual Basic, it was possible to fully exploit the graphical user interface technology; we have used multiple-document interface, object linking and embedding, dynamic data exchange, effective graphics and the possibilities to add custom controls by calling procedures in dynamic-link libraries.

Let us work through the *Woodstrat* system step by step: *Business Unit I* is one of the SBUs (the name and all the figures are fictitious); it is operating in several countries, with well-defined product groups and specified customer segments. Markets and segments differ for different product groups, and their importance varies over the planning period; the 1993 volumes and prices are used as starting points (the two small input boxes in the lower left hand corner, cf. figure 2) in order to calibrate the estimates to be given. The strategic Market Position (MP) is determined hierarchically; segments are defined for each product group and product groups are selected for each country; for each segment demand and price development estimates are made and consolidated to product group and country levels. The weighted average of the growth and price development estimates update the *Net sales* line in the income statement (cf. figure 8) through some functional links when the *CEO*-button is activated. In this way the market visions are formulated and immediately evaluated in terms of the income statement¹.

¹ cf the principles shown in fig. 1.

P. Walden and Ch. Carlsson

		Market	Position				
Position	<u>E</u> dit Grid <u>T</u> ools <u>H</u> elp						
Sweden Panel do Construc Sustai	ton industry 1 nable Competitive Ac	usines Ivantag	s Unit 1e —	1			CEO
		1994	1995	1996	1997	1998	ProdMix
Markets	Estim. Growth Devel.	4	5	3	4	4	
	Estim. Price Devel.	4	4	3	3	2	
ompetitors	Competitive Position	7.97	8.27	8.57	8.87	9.17	Print
	Critical Success Factors	3.45	3.50	3.34	3.28	3.50	
iwn Values	Competitive Position	7.30	7.60	7.90	8.20	8.50	
	Critical Success Factors	3.17	3.50	3.33	3.33	3.50	Calculate
	?Volume Development-%	774.5	5.9	2.8	-1.8	12.0	
1155	Volume Devel.(ton or m ^a)	10100	10700	11000	10800	12100	
	?Price Development-%	22.4	-1.7	2.8	-1.8	12.0	12
889	±Net Price Devel.(FIM)	1088	1070	1100	1080	1210	Adjust
	Total Sales (MFIM)	11.0	11.4	12.1	11.7	14.6	
							Exit

Figure 2: Market Position

The *Woodstrat* is supported with a fairly extensive database of country specific economic indicators and related forecasts, and market and segment specific forecasts on the development of price and demand levels. We found out that this helps the SBU managers to calibrate their assessments of growth and price developments - they do not have to guess or rely on some vague recollection of facts they happen to have acquired (cf. figure 3). With this model visions are anchored to facts about the strategic context (but price and demand estimates are not updated).



Figure 3: External Data

The Competitive Position (CP) is activated (with logical links) from the same level as the MP, and the MP and CP are worked out in parallel. The CP is determined in terms of critical success factors (these are SBU-specific and were determined successively in a series of seminars with the managers) by assessing the relative changes to the previous year, and as changes to the CP worked out for the previous year. This process is one of reassessing visions on MP when evaluated against critical success factors and the relative strengths of the competitors.

Comp	oetitive F	osition				
<u>P</u> osition <u>E</u> dit Grid <u>T</u> ools <u>H</u> elp						
Sweden Panel doors Construction industry	Busir	iess l	Unit	I		
Critical Success Factors	1994	1995	1996	1997	1998	
Quality Pricing	4	- 4	2	2	3	
Production Efficiency	1	2	3	4	4	
JOT	2	3	4	4	5	
Customer Orientation Service	5	5	4	3	3	
Marketing	4	4	4	4	3	
Planning	3	3	3	3	3	
CSF's average:	3.17	3.50	3.33	3.33	3.50	
Competitive Position Average: 7	7.30	7.60	7.90	8.20	8.50	
Competitor1 Competitor1	3.50	3.83	3.67	3.33	3.50	Calculate
Competitor2 Competitor2	3.67	3.33	3.17	3.33	3.50	
Competitor3 Competitor3	3.17	3.33	3.17	3.17	3.50	
CSF's average:	3.45	3.50	3.34	3.28	3.50	
Competitor 1 position: 8	8.30	8.60	8.90	9.20	9.50	Print
Competitor 2 position: 8	8.30	8.60	8.90	9.20	9.50	
Competitor 3 position: 7	7.30	7.60	7.90	8.20	8.50	
Competitive Position Average:	7.97	8.27	8.57	8.87	9.17	Exit

Figure 4: Competitive Position

146



Figure 5: Competitors

Three selected competitors are evaluated on the same critical success factors (CSF); as this is a benchmarking approach, quite a lot of time was used to select "good" competitors; CSF and CP averages were determined for the competitors; the relative difference in competitive positions is calculated and transferred with logical links to the CP, where it is used as a basis for assessing relative strategic CPs. There are functional links to MP, which are used to calculate an estimated development in volumes and prices; this function is only voluntary as the SBU managers can always make their own estimates and thus override the suggested development. This process anchors visions of relative CPs to facts (as they are known) about competitors.

A summary of relative strategic CPs and the expected development of the studied markets is shown in a summary graph (cf. fig. 6), which now is the first summarized, visual consequence of the MP and CP visions.



Figure 6: Market Graph

The third element, the Production Position $(PRO)^2$, estimates productivity as a consequence of the MPs and the CPs. *Production sold* is determined and transferred from the growth and price development specified in the MP; *productivity* is determined from several factors - labor, raw material, electricity, steam and technology; finally, there are lines for checking profitability and capacity limits (cf. figure 7). The productivity factors are updated from corresponding cost lines of the income statement (cf. figure 8) through functional, knowledge based links. The productivity and profitability measures are numerical consequences of key success factors of the MP and CP visions.

 $^{^{2}}$ cf. the principles shown in fig. 1.

s		Product	ion Pos	ition						
rodu	Business Uni	it I								
	1993 1994 1995 1996 1997 1998									
	± Production sold (milj. FIM)	103.25	84.68	92.90	96.07	93.79	113.99			
	Productivity index	100	64.06	69.81	74.57	72.88	85.65			
0.07	± Labor prod. index	100	79.91	83.44	85.34	84.49	91.88	L III		
0.20	± Raw material prod. index	100	.00	.00	.00	.00	.00	_ <u> </u>		
0.20	± Electricity prod. index	100	77.36	77.36	75.47	69.81	73.58			
0.20	± Steam prod. index	100	77.50	77.50	75.00	72.50	75.00			
D.33	± Technology prod. index	100	83.33	100.00	116.67	116.67	150.00			
	Prod. cost per ton or m ³	541	624	583	530	495	418	6.00		
	Profitability	2.09	1.83	2.06	2.29	2.41	3.19			
	Profitability index	100	87.56	98.56	109.57	115.31	152.63	P		
	Availab. Capacity (ton or m³)	100000	100000	100000	100000	100000	100000			
	Used Capacity (ton or m ³)	91214	74235	77522	79281	78493	85357	L _ A		
	Capacity Utilization	91%	74%	78%	79%	78%	85%	5		

Figure 7: Production Position

The CEO Report is activated from the summary level of MP. Functional, knowledge based links from the MP and a module for raw material costs update the revenue and cost lines of the projected income statement (cf. figure 8), which is linked with the balance sheet, the statement of funds and the key ratios (cf. figure 9); all these modules update each others with logical, knowledge based links in a way which follows proper accounting principles. Most of the lines are further specified in more details, but the summary reports are mostly sufficient for strategic planning purposes.

		CE	D-REPORT	ſ			
<u>File</u> Position <u>E</u> dit	tGrid <u>G</u> raph	Tools	<u>H</u> elp				
	Consolida	te Calculat					
		Print	Exit				
INCOME STATE	MENT 🛃	1993	1994	1995	1996	1997	1998
± Net Sales		73.1	84.7	92.9	96.1	93.8	114.0
Change in Inventory (finished goods)	7.8	0.0	0.0	0.0	0.0	۵0
Other Income		16.7	16.7	19.0	20.0	21.0	22.0
± Variable Costs		35.7	40.9	43.0	45.3	47.6	49.9
± Fixed Costs		6.1	6.8	7.6	8.1	8.7	9.2
Operating Margin		55.8	53.7	61.3	62.6	58.5	76.9
Operating Margin - %	Ď	76.3	63.4	66.0	65.2	62.4	67.4
± Depreciation		16.3	17.7	17.7	17.7	17.7	17.7
Operating Profit		39.5	36.0	43.6	44.9	40.8	59.2
Operating Profit - %		54.0	42.5	46.9	46.7	43.5	51.9
± Financial Items Net		-18.9	-14.0	-12.0	-8.0	-4.0	3.0
Profit after fin. items	-	20.6	22.0	31.6	36.9	36.8	62.2
Extraordinary income	& expenses	-3.2	-1.9	-4.0	-12.0	-23.0	-30.0
Result		17.4	20.1	27.6	24.9	13.8	32.2
	Click on panel to	choose uni	t				

Figure 8: Income Statement

The Income Statement is enhanced with several specified reports, of which only the report on Key Ratios is shown here (cf. figure 9):

P. Walden and Ch. Carlsson

ile	<u>P</u> osition	<u>E</u> dit Grid	<u>G</u> raph	Tools	<u>H</u> elp				
Business Unit I									te Calci
KE	Y RATIO	3	±	1993	1994	1995	1996	1997	1998
RO	NA-%			16.7	16.3	21.1	23.4	26.3	42.2
Net	Gearing			727 A	343.1	160.1	78.8	42.0	low
Cap	ital Turnove	r Rate		0.3	0.4	0.4	0.5	ð.0	£0
We	rking Capita	d Turnover R	ate	1.9	2.5	2.7	2.8	2.7	32
Equ	ity Ratio-%			10.9	19.9	32.9	46.7	63.7	85.2
Ret	urn On Equi	ty		60.4	41.1	36.1	24.6	12.0	21.8
Inve	estment-% o	f Net Sales		4.5	0.0	0.0	0.0	0.0	0.0
Rat	e of Self fina	ncing Innects	nents	84.8	0.0	0.0	0.0	0.0	10

Figure 9: Key Ratios

The main benefit with the automatic linking of the modules of the CEO Report is that the managers do not have to interrupt their work on strategic assessments in order to check on the profitability and the financing - a task which is hard on non-specialists. The links have also turned out to be major time-savers.

As Return on Net Assets (RONA) is a forestry industry standard, a graphical simulation module for RONA allows the managers to quickly find critical sales or operating cost levels for reaching target levels of the RONA (cf. figure 10):



Figure 10: RONA Simulation

A feature which won much approval among the SBU-managers were the integrated modules for simulating investment alternatives and corresponding alternative financial models, which showed their impacts on cash flows and key ratios (cf. figures 11 and 12).

Pro	cess In	vestme	nts			-
<u>F</u> ools						
Ad hoc investment	usin	ess U	nit l			
Investment Plan						
	1994	1995	1996	1997	1998	Finance
±Impact on sales (milj. FIM)		7.86	16.06	23.82	34.52	
Total cost for investment		51.2	10.6	9.9	9.3	
±Impact on variable costs		0.0	0.0	0.0	0.0	
±Impact on fixed costs		0.0	0.0	0.0	0.0	Graph
Total impact of investment		-43.3	5.5	13.9	25.2	
Accum. impact of investment		-43.3	-37.8	-23.9	1.3	
Average sales increase		20.6	20.6	20.6	20.6	L GRUEBARG
Average cost increase		20.0	20.0	20.0	20.0	
Average return		0.6	0.6	0.6	0.6	
						Print
Impact on volume (m ³ , ton)	100	100	100	100	100	
Impact on price (FIM)	100	100	100	100	100	
Beginning year 1995	F	rimary in	vest. (mi	lj FIM)	80	

Figure 11: Investment Plan

-	Process Investments						·
Tools							
Inv. Fir	Ad hoc investment Ad hoc investment ancing Plan	usin	ess U	nit l			
		1994	1995	1996	1997	1998	Inv.Plan
	Share of own fin. (milj. FIM)		40.0	0.0	0.0	0.0	
	Payment to interest		3.2	2.6	1.9	1.3	
	Payment to principal		8.0	8.0	8.0	8.0] .
	Total payment amount		51.2	10.6	9.9	9.3	Graph
	Accum. payment to principal Outstanding principal		8.0 32.0	16.0 24.0	24.0 16.0	32.0 8.0	Colculate
	Depreciation		13.3	13.3	13.3	13.3	
	Investment after depr.		66.7	53.4	40.1	26.8	
	Share of loan financing % 50 Share of own financing % 50 Annual interest rate % 8	Loan life (years) Depreciation years				5 6	Print

Figure 12: Financing Plan

Woodstrat is used for strategy formation, which was seen in the use of the various modules:

- i. the database of *external data* served as an instrument to establish reference points for growth and price estimates;
- ii. the *MP* module served both analysis and synthesis; it was first used to build estimates of price and volume developments for specific product groups in specific market segments;

then it was used to get a feeling for the expected development in a product group or in a country; then the estimates were reiterated until some acceptable levels were reached;

- iii. the *CP* module was used for similar analysis / synthesis iterations in the same countries, product groups and market segments but the iterations were now in terms of critical success factors; reference points for these were established with analogous iterations of the critical success factors for three key competitors; the results were used to establish relative competitive positions; these positions were then used to establish a new set of reference points for the estimates of price and volume developments;
- iv. the price and volume estimates were reiterated with the proposed product mix (PM) combinations, which are essential for estimating the use of raw material (a major cost factor) and production capacity;
- v. the MP, CP and PM modules are then used to determine the net sales and variable costs of the income statement (PROF) in the *CEO* module; when worked out with a complete set of variable and fixed costs the resulting profit is either satisfactory or non-satisfactory; in the latter case it triggers reiterations of the MP, CP and PM modules; if satisfactory it is further evaluated in terms of the capital structure, the use of funds and key ratios (PROF); reiterations can be triggered on a number of reference points;
- vi. the CP determines the need for investments; the corresponding module triggers the financing of these investments; reiterations are triggered by reference points in the income statement, the capital structure, the use of funds and key ratios;
- vii. the MP and the CEO modules update and trigger the determination of the productivity index; reference points induce reiterations of investments and of the MP and CP modules;

There are a number of minor processes also included in the system, but the major processes shown in (i) - (vii) represent the strategy formation process.

Experiences of Strategy Formation with Woodstrat.

The *Woodstrat* was run in 9 SBUs in a series of seminars in the Spring of 1994^3 in sessions which involved about 40 managers in a 2-month period and culminated in presentations given by the COOs of the SBUs to the corporate board. In these presentations they used *Woodstrat* to demonstrate their strategic solutions and to show the internal logic of their plans. As we had included an on-line *Memo* function in the system and charged the mangers with continuously commenting on their assessments and conclusions in all modules, we could follow their reasoning. The most effective management team collected 15 pages of information in their Memo.

The MP module helped in determining the expected volume and price development in key market segments. The managers were able to determine their expected market development in more detail than ever before, and also found out that they now have a basis for quick and systematic evaluations of unexpected changes in both volumes and prices. The CP module was one of the key modules from the very beginning. In most SBUs it was found out that: (i) the management team does not know their key competitors as well as they would like to know them; (ii) they are not too well aware of what their critical success factors are, and what impact various levels of these might have on their success in various market segments, and (iii) they were not too sure about how they could improve on these factors or how this improvement would work out in a chosen planning interval. The CP module offered a possibility to work out all implications of the various factors systematically, and it offered a good basis for explaining their competitive solutions and defending their requirements for investments in a manner much more persuasive than before. The CP module required quite a lot of work, which felt tedious from time to time, but it was probably the most rewarding of all the modules.

The hyperknowledge environment made the evaluations of relative CPs fast and efficient. A summary of the comments showed the basic problem to be the lack of sufficient knowledge of the competitors (which in our mind was surprisingly good). The gradual assessment of relative competitive positions worked very well despite the fact that the assessments were basically analytical; the supporting discussion in the Memo showed systematic and consistent reasoning about (i) the abilities of the competitors and (ii) the possibilities to develop critical success factors through investments and increased productivity.

³ The system has now been used in an additional 8 SBUs with more than 40 managers participating.

P. Walden and Ch. Carlsson

The users were quite familiar with building the elements of the CEO Report (PROF) as this part of the strategy formation process has been routine for a number of years. The contribution of *Woodstrat* was that the knowledge based links made operations more precise and much faster than before; all required operations were fully automated and a good selection of graphics objects helped to produce effective and useful summary reports.

The investments module (INV), with an integrated financing module (FIN), was built to help the SBU managers work out their investment proposals. It was found to save quite a lot of time as complex and detailed investment plans, with a number of alternatives scheduled for a number of years, and their impact on financing, key ratios, capital structure, etc. could be evaluated.

The investments module, and its interlinked financing module, was technically more advanced than the other modules; it seems that investment analysis was intuitively understood by only a select few, but that *Woodstrat* made the logic quite for all SBU managers.

At some point we almost believed that the determination of *productivity* is a form of black magic, as nobody seems to be able to come up with a reasonable definition. Finally, we built and tried out our own version of the productivity measures, and they seemed to work. Productivity is an abstract and complex measure, and the idea was to gain some insight into the effects of the MP and CP visions of changes in productivity.

There are knowledge based links to simplify the calculations, and (as we found in the Memos) the users turned out to gain an intuitive understanding of the relations between market developments and the productivity.

The External data module was built to be easily accessible and to give a benchmarking basis for estimates of the expected development of demand and prices. It worked basically as expected, but there was a further demand for better depth of knowledge and more industry-specific information. The hyperknowledge environment with quick access to data, and the possibility to browse back and forth between various databases proved to be timesaving and to promote the benchmarking processes we wanted to get started.

The Memo module turned out to be an exciting tool for us, the system developers, as it displayed all the insights the users gained in their discussions of the various alternatives. It also showed their conclusions on various items and their evaluation of their competitors. A number of factors were not known too well and the Memo was used to register those points were follow-up studies of the competitors were required; a number of questions and ideas were forwarded to sales offices in Europe for verification and follow up studies. The Memo revealed that we were

154 More Effective Strategic Management with Hyperknowledge: Case Woodstrat

following *a strategy formation process*, which was carried out by the management teams of the SBUs.

Conclusions

Woodstrat is a hybrid of an object-oriented expert system and a hyperknowledge support system, and is thus constructed to provide both advanced level knowledge support in strategic management and an environment for linking assessments of qualitative factors with systematic quantitative evaluations of their consequences. We found out in the *Woodstrat* project that hyperknowledge can be created with object-oriented visual tools, which represent a new software technology for KBS construction. The objects can be given semi-expert properties and then be linked with the type of relations that form the hyperknowledge environment.

Woodstrat is a support system for strategy formation. The links between the logical elements of the system follow an intuitive, internal logic which has gradually emerged through interactive work with the SBU-managers. This has created the foundations for a quick and effective user acceptance.

Woodstrat appears to have a number of useful features as reported by the users:

- the system guides the user to focus on important issues which eliminates unnecessary work;
- although I have for several years done strategic planning "my way", after having used the Woodstrat system I would not change it for anything;
- I am very pleased with the system, it really is a working system; although I also have (paper) documents of the SBU plans, I work only with the Woodstrat system; I have got very positive feedback from the CEO on the computer-based presentation of the division's and the SBUs' strategic plans; perhaps I ought to allow the SBUs to carry out group and division consolidations this I believe would generate internal competition to attain divisional goals and objectives;
- Woodstrat compared to "my old way"? I worked more thoroughly and I used more time than normally; next year the work will be accomplished in much less time as I now have a fairly complete basis as a starting point;
- the final version which we presented to the corporate executive board took us 1,5 day to finish (preparatory work done); our group six people worked as a team and found the system very useful for teamwork;
- the support system "imprisoned us; the drawback was that we concentrated too much on details in the MP and CP;

- we had our basic visions and missions in mind, and found out through *Woodstrat* that our visions of chlorine free pulp changed; there exists after all a demand big enough for the new product we had discussed;
- the planning process was real team work;

The *Woodstrat* system was built as a series of interactive prototypes, with the eventual users being part of the design process from the beginning. In the *Woodstrat* project we were able to get even senior managers to become active systems users, as they found out that they could use the KBS technology to formulate their own perceptions of a strategic context.

References

- 1. **Rowe**, Alan J. and Boulgarides, James D. Managerial Decision making, Macmillan Publishing Company, New York (1992)
- 2. Day, George, Weitz, Barton and Wensley, Robin, (ed.) *The Interface of Marketing and Strategy*, Jay Press Inc., Greenwich (1990)
- 3. **Mintzberg**, Henry *The Rise and Fall of Strategic Planning*, Prentice Hall (1994)
- 4. Ansoff, H. Igor Critique of Henry Mintzberg's 'The Design School: Reconsidering the Basic Premises of Strategic Management', *Strategic Management Journal*, Vol 12 (1991), 449-461
- 5. **Mintzberg**, Henry Patterns in Strategy Formation, *Management Science*, Vol 24, No. 9, (1978)
- 6. Eden, Colin Strategy Development and Implementation-Cognitive Mapping for Group Support, in Hendry and Johnson (eds): *Strategic Thinking: Leadership and the Management of Change*, (1993)
- 7. Carlsson, Christer Expert Systems as Conceptual Frameworks and Management Support Systems for Strategic Management, *International Journal of Information Resource Management*, Vol 2, No. 4, (1991), 14-24
- 8. Carlsson, Christer New Instruments for Management Research, Human Systems Management, Vol 10, No. 3 (1991), 203-220
- 9. Carlsson, Christer Knowledge Formation in Strategic Management, Proceedings of the HICSS-27 Conference, 1994, 221-240
- 10. Carlsson, Christer and Walden, Pirkko Strategic Management with a Hyperknowledge Support System, *Proceedings of the HICSS-27 Conference*, 1994, 241-250

- 11. Carlsson, Christer and Walden, Pirkko Cognitive Maps and a Hyperknowledge Support System in Strategic Management, *Group Decision and Negotiation*, (1995, forthcoming)
- Carlsson, Christer and Walden, Pirkko On Fuzzy Hyperknowledge Support Systems, NGIT'95 Proceedings, Tel Aviv 1995
- 13. Walden, Pirkko and Carlsson, Christer Enhancing Strategic Market Management with Knowledge Based Systems, *HICSS-26 Proceedings*, IEEE Computer Society Press, Los Alamitos, 240-248
- 14. Walden, Pirkko and Carlsson, Christer Strategic Management with a Hyperknowledge Support System, *HICSS-27 Proceedings*, IEEE Computer Society Press, Los Alamitos, 241-250
- 15. Walden, Pirkko and Carlsson, Christer Hyperknowledge and Expert Systems: A Case Study of Knowledge Formation Processes, *HICSS-28 Proceedings*, IEEE Computer Society Press, Los Alamitos (1994)
- 16.Chang, Ai-Mei, Holsapple, Clyde W. and Whinston, Andrew B., Model Management Issues and Directions, *Decision Support Systems*, Vol 9 (1993), 19-37
- 17.Gershman, Anatole and Gottsman, Edward Use of Hypermedia for Corporate Knowledge Dissemination, *HICSS-26 Proceedings*, IEEE Computer Society Press, Los Alamitos, 411-420
- 18.Lange, Danny B. Object-Oriented Hypermodeling of Hypertext Supported Information Systems, *HICSS-26 Proceedings*, IEEE Computer Society Press, Los Alamitos, 380-389
- 19.Bonczek, R.H., Holsapple, C.W. and Whinston, A.B. Foundations of Decision Support Systems, Academic Press, New York 1981

III. Research Notes: Using DSSs

Learning Decision Making through Management Games

Timo Leino

Turku School of Economics and Business Administration P.O. Box 110, 20510 Turku, Finland

This paper deals with the use and development of management games. A management game is a session where the groups of players form fictive companies to operate at fictive markets. The market operations are carried out by using a special computer software. The objective is to enhance the players' decision making skills and the insight of the industry. Management games are used mainly in teaching purposes, but they may also be appropriate planning tools when used as simulation or what-if models.

In a management game a number of competitive (long-time) strategies may be pursued alongside with various tactical and operative short-time decisions regarding for instance personnel, equipment investments, financial operations or marketing campaigns. Therefore, the development team of a management game software must integrate the theoretical knowledge of market economies with the practical know-how of the chosen market environment. The development process is somewhat similar to those of expert systems, combining the knowledge of human experts and text books with system engineering skills.

A successful game session is a harmonised combination of motivated players, a suitable and carefully planned setting, an experienced game instructor and, finally, advanced software. <u>The</u> Shipping Game[®] is a state-of-the-art computer-aided management game for the shipping industry. It incorporates the latest advances in software development for personal computers with theoretical knowledge in maritime economics and practical know-how in shipping into a thrilling dynamic liner shipping environment.

The goal in designing and conducting a game session must be clear. There are several alternatives one may regard as important objectives, such as pedagogic objectives, increased team work abilities and increased knowledge. <u>The</u> Shipping Game[®] may be used as a vehicle in teaching decision-making in different competitive settings, liner shipping operations management and the use of economical indicators supporting decision making.

Management gameS

A management game is a session where the groups of players form fictive companies to operate at fictive markets. The market operations are carried out by using a special computer software. The objective is to enhance the players' decision making skills and the insight of the industry. Management games are used mainly in teaching purposes, but they may also be appropriate planning tools when used as simulation or what-if models.

There is a wide spectrum of management games according to the objectives that are set to the game. Figure 1 illustrates the typology. First, there are general purpose games simulating the whole operative environment of a firm and functional games focusing on a special type of decisions, e.g. marketing or finance. Second, some real life industry may be used as the market environment, or the markets are defined in more general terms. The third dimension concerns the competitive situation: the firms are playing against each other or against anonymous competitors, i.e. the computer. In the former case the decisions are compared to those of the competitors while in the latter case they are compared to predefined optimal decisions.



Figure 1. The typology of management games.

An elegant example: <u>The</u> Shipping Game[®]

<u>The</u> Shipping Game[®] is a state-of-the-art computer-aided management game for the shipping industry. It incorporates the latest advances in software development for personal computers with theoretical knowledge in maritime economics and practical know-how in shipping into a thrilling dynamic liner shipping environment.

The purpose of <u>The</u> Shipping Game[®] is to enhance the management skills of the players in operating a liner shipping company in a close-to-real market environment. In the dynamic markets a number of competitive strategies may be pursued

alongside with various tactical and operative decisions regarding, among other things, fleet employment, route planning and personnel management.

In a typical game situation you need one PC for the Game instructor and one PC for every team participating the game. You can run software in any personal computer using Windows 3.1 preferably with a 486+ processor and 8 MB of RAM. The Shipping Game[®] is designed to be used in stand-alone workstations without benefiting from LAN systems, and, on the other hand, avoiding the possible problems caused by exotic network configurations. Therefore, all data transfer in <u>The</u> Shipping Game[®] is meant to be managed by using diskettes. To run the software you need Microsoft Excel 4.0 and the Run-only version of Level 5 Object 2.5, the latter only in instructor's PC and the former both in instructor's and players' PC's.

Building Management games

Naturally, the structure and functions of a management game software depend on the aimed use of the game. However, some core functions are needed in every game. These include (see Figure 2):

- A program with which the players plan and enter their decisions. It must be user-friendly and assist the players in routine tasks (calculations) but may not give straight guidelines to decisions. Decision making is the task that the players should learn and therefore it must be left totally to them. In <u>The</u> Shipping Game[®] this program is designed in Excel 4, which is a tool with features needed to create a modern, Windows-based application that is easy to use and nice to watch.
- A program with which the players may browse the market data and the reports of their own activities, such as statement of income, balance sheet and sales reports. This may be apart from the decision making module or one function of that module. Use of graphics is preferable to show trends, variation or comparisons. Using graphs is one thing the players may learn through a management game. In <u>The</u> Shipping Game[®] this program is also designed in Excel 4; spreadsheet is with no doubt the best tool in producing reports.
- The "game" itself, that is, a program that simulates the markets. This is the soul of a game software. In a competitive setting (a game against others) the game instructor operates and adjusts this program. Therefore, the requirements for user-friendliness are not so high, more important attributes are efficiency, robustness and reliability. In <u>The</u> Shipping Game[®] this program is designed by

using Level 5 Object v2.5, which is a Windows-based, objectoriented expert systems generator. It showed not to be a very efficient tool in this project, because of low efficiency and some problems in combining rules, procedural code and external batch files. If we could start again from scratch, we would probably choose Visual Basic instead.

• Programs to transfer the data between teams and the game module. These programs are important because the actions in data transfer are the most error-proning. On the other hand, the task is only to move or copy files, which makes the programs quite easy to construct. In <u>The</u> Shipping Game[®] the data transfer is taken care of by simple batch files, which are very easily and quickly modified in the case of some unusual technical configurations in a new game environment.



Figure 2. A general structure of a management game software.

There are lot of alternatives in modelling the markets, but the following questions have to be answered:

• O How the demand is created? The most simple way is to predefine it as a line of numbers indicating the demand in each period and market area. In more advanced solutions the demand is reactive, depending on the actions at the markets. In <u>The</u> Shipping Game[®] the demand is defined for each period and route using an algorithm that uses as input the changes in service level and price level of the actors at the markets, the competitive situation (monopoly / oligopoly / competition) and the type of the customers (big / small).

- How the competitors are evaluated? The offers of the competitors must be evaluated by using several criteria, resulting in one or several image indicators for each supplier, indicating the motivation of the customers to buy from that particular supplier. The choice, number and weights of the criteria is a complex problem and solving it requires expertise. Text books give general guidelines, but if the game is tailored for a special line of business the general knowledge is not enough. In <u>The</u> Shipping Game[®] the firms are offering sea transport services. The criteria are the price (freight), the service level (quality) and the transport ability. These are calculated through a quite complex process.
- O How the concrete sales volumes are defined? There are two main solutions. One possibility is to rank the suppliers according to their image indicators and share the demand according to some predefined rule, for instance 70% to the best one (if they can deliver that much), 20 % to the second, etc. This solution is probably the best one if the game simulates business-to-business markets where the number of customers is very low. Another possibility is to calculate the market shares directly from the image indicators. This fits better to a situation where the number of customers is larger and the transactions are not based on long-term partnerships. In any case, the inflexibility of markets has to be taken into account to avoid too dramatic changes in competitive positions.

A Management game session

A successful game session is a harmonised combination of motivated players, a suitable and carefully planned setting, an experienced instructor and, finally, advanced software like <u>The Shipping Game</u>[®].

From the game instructor's viewpoint a game session has four phases as follows:

1. Preliminary operations. Before the game the instructor has to plan the game setting, adjust the parameters of the software, and prepare the teaching environment, e.g. install the software.

- 2. Introduction. The game starts with a short lesson, where the instructor leads the players into the game environment and decisions of the game objectives are made.
- 3. Playing the game. The game itself takes usually several hours and may be conducted in a number of ways, depending on the objectives. In most games the time is managed by using periods, the length of which is for instance one year or one month. The teams make the operative decisions always for one time period, preferably basing them on some long term plan.
- 4. Summary. A discussion at the end is important, the game should be analysed and the key points summarised from the learning perspective.

In the following, some guidelines for planning and conducting a game session are given.

- The time available for the game. Generally, with first-time players, ten hours of effective playtime or more is needed. Hence, a two-day session is recommended.
- Skills of the players. Prior knowledge in business decision making may reduce the time needed. Alternatively, it may give more possibilities to vary the game setting. The skills must be analysed in advance.
- The set-up situation at the beginning. Various starting points may be used according to the level of the players and the objectives of the game. The more experienced players, the more complex setup may be chosen.
- Equipment at hand. It's very critical to check in advance the technical details, i.e. smooth functioning of all the modules of the management game software with the computers and printers reserved for the game session.

A careful planning of the game is necessary. Usually the software allows several alternatives to start and lead the game. As an example, in <u>The</u> Shipping Game[®] there are 7 market areas (e.g. Germany Baltic Ports) available between which the vessels may be routed. Thus, if all 7 areas are available, there are 7 x 6 = 42 different routes (markets) the firms may choose. In the beginning of a game that is too much, the players are not able to manage so many markets effectively. Usually we begin with only 2 areas, which gives only 2 routes, and then increase the number of markets when the game proceeds.

T. Leino

The key issue in achieving the game goals is an analysis of the players' skills and a suitable game setting based on that analysis. The prior skills and knowledge of the players must match the game's level of difficulty. Too many features or too many details may be confusing and limit the learning of essential matters. On the other hand, too simple setting may hamper the game being challenging or close-to-real and lower the motivation.

The effects to decision making skills

Designing and conducting a game depends on what you want to achieve through the game. Clear objectives are necessary to plan the game setting in a sound way. Instructor's behaviour and actions during the game must also follow from these objectives. There are several alternatives one may regard as important objectives, such as pedagogic objectives, increased team work abilities and increased knowledge. The Shipping Game® may be used as a vehicle in teaching

- O liner shipping operations management,
- O decision-making in different competitive settings,
- O use of information to support decision-making, and
- O business knowledge, e.g. general accounting principles.

Depending on the objectives the game instructor should during the session take up various topics to be analysed and discussed. The session surely goes to a wrong direction without strong leading: it becomes nothing but a game without any learning effects. The players must be guided in using the information available, otherwise they will make bad decisions which result in losses. So, the game instructor plays a very important role.

In a management game there usually are lot of data available - as in real business. The essential thing to learn is how to analyse the mass of numbers and find the critical ones. Also, the most needed figures are not ready but must be produced by the players themselves. In good case, through a management game the players really learn how to use the numbers to make better decisions. Very often the "trial and error method" is used, the firms face problems and their task is to solve them and make the firm succeed. In a competitive situation (a game against others) every group does not succeed, but that belongs to the game.
Putting models to work. A comparison between user- and research driven projects

Karin Mossberg

National Defence Research Establishment, Department of Defence Analysis, S-172 90 STOCKHOLM, Sweden

Getting advanced computer models accepted as instruments of analysis and learning can be an intricate task. Our experience indicates that the acceptance of the models depends on the active participation of the receiving organisation in the development and implementation process. On the other hand, some of the objectives of an advanced model may not be achieved with the users as the major driving force.

In this paper, two models and the process of their development and implementation with the Swedish Navy will be discussed and compared. The first model, a distributed system for simulation of anti-submarine warfare, was initiated at our research institute and is now used by the Navy as a development tool for new tactics. The other model, displaying registered ship movements, is an instrument for analysing military manoeuvres. This model had its origin in a request from the Navy and has successively been modified to meet new fields of applications.

Keywords: military application, development, implementation, models

Operation research in the Swedish Navy

The work described in this paper has been performed by a group that works towards the Naval Warfare Centre in the Swedish Navy. The role of this group is to act as consultants in the incorporation of operation research in the development and evaluation of tactics. When a tactical concept is to be developed or modified, the Navy forms a group consisting of officers and one or two researchers from our group. The officers have the military experience and are also the persons that will use the tactic in a later stage. Our obligation is to ensure that the creative process is done in a well-reasoned and logical way and that all relevant factors are considered. It is also our responsibility to make calculations, handle computer models and to make evaluations when necessary.

One of our main fields the last years has been anti-submarine warfare tactics. Since the Swedish Navy operates in the Baltic, which has an archipelago and shallow water that differs substantially from that in the Ocean, it requires a unique thinking in anti-submarine warfare questions. The military scenario in the surrounding world changes from year to year and so does the technical equipment of the warships. Consequently, there is a continuos need for modification of the tactical concepts.

In this work, computer models for simulation, analysis and learning are often needed, and it is mostly our responsibility to develop and evaluate these models. In most cases it is preferable that the officers use the models, if necessary with our guidance, since it is their experience that should form the input to the model. They are also the ones that should use the result when the study is ready. By that reason I will in this paper call the officers the "users" and our group the "researchers".

Two models will be discussed in this paper, one that was initiated by our group and one that had its origin in a request from the Navy. The differences in their development- and implementation processes will be discussed below and some conclusions will be drawn.

The research driven project -

A distributed system for simulation of anti-submarine warfare

The process of tactic development includes activities such wargaming, training in simulation establishments and manoeuvres at sea. When we started to work with these questions, our group was lacking a level where to play more simple games focused on the tactical concepts without being caught into problems such commanding or sonar skill. We wanted to have a simulation model where the computer made the decision if the warships had contact with the submarine or not. The idea was also that it should be easy to use the model for one or a group of persons that hadn't much time for preparations.

Our group initiated the development of this model a few years ago and modifications are still undertaken. The model allows several people to play the game on different computers. All participants have the same map on the screen and they can chose if they want to see the other participants or not. The game is updated with a time interval between which the ships can change direction, speed and parameters of the sonar's. The computer displays the ships positions and the sonar contacts both for the ships and the submarine.

The result of the simulation largely depends on the experience of the officers and it is therefore of great importance that the officers are involved and engaged in the simulations. Our intention was that we in the beginning should use the model together with the officers and that they after some period should be able to use it on their own.

K. Mossberg

This model was driven through by us, the officers were only marginally involved during the development. They where participating first when the model was almost ready and it was time to do the first test simulations. At that stage most of them had a lot of viewpoints on the model and wishes about other features that should be implemented. The most frequent criticism was that they wanted a similar model but with another purpose, more aimed at, for example, geographical studies or sonar training. We then had to argue for our point of view and in some cases to implement their wishes in the model. The entire process of implementation therefore took a long time, going on for several months.

The user driven project -

An analysis instrument for evaluating of manoeuvres

During the spring -94 we were asked to help the Navy to evaluate some of their manoeuvres when training anti submarine warfare in the archipelago. At that time they were lacking an analysis instrument to be used during the manoeuvre to help them to rapidly take decisions about where to search for the submarine after that they had had a short contact. We were asked to have viewpoints on such an instrument and later on we took the responsibility of the realisation of such a computer model. Before we started to programme the model we had several discussions with the officers guiding us to which features that should be included in the model.

The model, which now is completed, shows a map of the area where the manoeuvre takes place. It also shows the positions of the ships that have been registered automatically by GPS as a function of time. Besides the ship's positions it is possible to show the ranges of the sonar's, the area that has been searched for, submarine contacts and sonar buoy positions. From that information and from assumptions on the speed of the submarine, the model gives an apprehension of where the submarine may or may not be at the moment.

In this case, there were no problems to get the model accepted. The users were involved in the development from the beginning and we had frequent discussions during the development. When the model was ready, it was very well received and it was also used for other purposes than was thought of from the beginning, wherefore new features were implemented afterwards. The entire process from the first idea until the model was ready and distributed took no more than half a year.

Conclusions

There are major advantages when the project is initiated by the user. The user has then a large interest in the model from the beginning and is naturally involved in the development process. Thereby, it will probably be no problem to get the model accepted when it is ready. At the same time, it is possible for the researcher to influence the contents of the model.

But models are not always initiated by the users. In our position, having the responsibility for that a study is driven in a correct way, it is our obligation to take the initiative to new methods and computer models when needed. If one's intention is to teach a new way of thinking, a computer model can be a good instrument for that purpose. It could be done, for example, by letting the computer force the user to take decisions in a given sequence that differs from the way by routine. In such cases it is a necessity that the model is initiated and driven by those persons who want to teach something, which in our case often is the researcher.

Whatever the reason is for the researcher to initiate a new model, our experience tells us that it is extremely important to make the participation of the receiver as active as possible in an early stage. It is also necessary to present the model in a way that convince the user about it's justification and to deposit time for that process.

It must also be remembered that there are lots of other factors that will influence the acceptance, such as if the model is small or comprehensive, if it acts as a black box or is transparent to the user, which platform (Mac/PC) it is implemented on, the interface to the user and the amount of data that will be needed as input to the model..