INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
Mental models for strategic management: Representation and inference in a management support system

Carlson, David Allen, Ph.D.
The University of Arizona, 1991

Copyright ©1991 by Carlson, David Allen. All rights reserved.
MENTAL MODELS FOR STRATEGIC MANAGEMENT: REPRESENTATION AND INFERENCE IN A MANAGEMENT SUPPORT SYSTEM

by

David Allen Carlson

Copyright © David Allen Carlson 1991

A Dissertation Submitted to the Faculty of the COMMITTEE ON BUSINESS ADMINISTRATION
In Partial Fulfillment of the Requirements For the Degree of DOCTOR OF PHILOSOPHY In the Graduate College THE UNIVERSITY OF ARIZONA

1991
As members of the Final Examination Committee, we certify that we have read the dissertation prepared by David Allen Carlson entitled Mental Models for Strategic Management: Representation and Inference in a Management Support System and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

Dr. Sudha Ram

Dr. Barbara Gutek

Dr. Jay F. Nunamaker, Jr.

Dr. Alvin Goldman

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copy of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

Dissertation-Director

Dr. Sudha Ram

June 14, 1991
STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under the rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgement of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the copyright holder.

SIGNED: ____________________________
ACKNOWLEDGMENTS

I would like to thank my committee members--Sudha Ram, Barbara Gutek, Jay Nunamaker, Alvin Goldman, and Merrill Garrett--for their academic support and encouragement during this research project. I especially wish to thank Sudha Ram for guiding my efforts to publish our research results. I also offer sincere thanks to Alvin Goldman for his patient instruction in the philosophical foundations of my work and for helping me to appreciate the theoretical perspective.

Special thanks to Mary Baroni for originally suggesting that I obtain my doctorate and to Scott McIntyre for introducing me to artificial intelligence and for guiding me along the way. Although I have occasionally doubted the wisdom of them both for pulling me onto the academic treadmill, I am very grateful now that I have finished this first step.

Most of all, I would like to dedicate this dissertation to my parents, Art and Donna Carlson, for their constant encouragement in my studies and for their outstanding example of lifelong learning.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF ILLUSTRATIONS</td>
<td>11</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>13</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>14</td>
</tr>
<tr>
<td>CHAPTER 1: INTRODUCTION TO MENTAL MODELS</td>
<td>16</td>
</tr>
<tr>
<td>What are Mental Models?</td>
<td>17</td>
</tr>
<tr>
<td>The Strategic Management Process</td>
<td>21</td>
</tr>
<tr>
<td>Field Research and Computational Models</td>
<td>22</td>
</tr>
<tr>
<td>CHAPTER 2: A THEORETICAL FRAMEWORK FOR MENTAL MODELS</td>
<td>28</td>
</tr>
<tr>
<td>The Mental Model Concept</td>
<td>29</td>
</tr>
<tr>
<td>Defining Mental Models</td>
<td>30</td>
</tr>
<tr>
<td>Managerial Problem Solving</td>
<td>33</td>
</tr>
<tr>
<td>The Nature of Mental Models</td>
<td>37</td>
</tr>
<tr>
<td>Models in the Mind</td>
<td>38</td>
</tr>
<tr>
<td>Table of Contents (continued)</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Mental Representation</td>
<td>40</td>
</tr>
<tr>
<td>Mental Simulation</td>
<td>41</td>
</tr>
<tr>
<td>Organizational Psychology</td>
<td>44</td>
</tr>
<tr>
<td>Cognitive Complexity Theory</td>
<td>46</td>
</tr>
<tr>
<td>Complexity-Theoretic Propositional Framework</td>
<td>51</td>
</tr>
<tr>
<td>Image Theory</td>
<td>56</td>
</tr>
<tr>
<td>Complementary theories</td>
<td>62</td>
</tr>
<tr>
<td>Summary</td>
<td>62</td>
</tr>
</tbody>
</table>

**CHAPTER 3: ESTABLISHING THE NATURE OF MENTAL MODELS**  65

- Research Design  67
- Design for the Case Study Research  69
  - Organizations Studied  70
  - Multiple-case, Embedded Design  71
- Data Collection  73
- Data Analysis  74

**CHAPTER 4: STUDIES IN THE WAY OF MENTAL MODELS**  77

- The Hoshin Management Process  78
- Alpha-Tech Corporate Headquarters  78
- Alpha-Tech Product Division  80
<table>
<thead>
<tr>
<th>TABLE OF CONTENTS (continued)</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Spirit of Hoshin</td>
<td>80</td>
</tr>
<tr>
<td>Total Quality Management Strategy</td>
<td>83</td>
</tr>
<tr>
<td>Malcolm Baldrige Criteria for TQM</td>
<td>84</td>
</tr>
<tr>
<td>Managing a TQM Strategy</td>
<td>84</td>
</tr>
<tr>
<td>A Model for Organizational Productivity</td>
<td>86</td>
</tr>
<tr>
<td>Background</td>
<td>87</td>
</tr>
<tr>
<td>Model Structure</td>
<td>89</td>
</tr>
<tr>
<td>Decision Heuristics</td>
<td>93</td>
</tr>
<tr>
<td>Mental Simulations</td>
<td>95</td>
</tr>
<tr>
<td>Summary</td>
<td>96</td>
</tr>
</tbody>
</table>

CHAPTER 5: REPRESENTING MENTAL MODELS .......................... 98

A Functional Architecture for Mental Models .................. 101

Representational Propositions for Mental Models .......... 105

The Structure of Complex Knowledge .......................... 107

Complexity and Strategy Formation ...................... 117

A Computational Architecture ............................. 120

Linguistic Layer .................................. 121

Epistemological Layer ............................ 121

Object Management Layer .......................... 124

Implementation Layer .............................. 125
**TABLE OF CONTENTS (continued)**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schemata, Frames, and Inference</td>
<td>125</td>
</tr>
<tr>
<td>An Overview of Frame Representation</td>
<td>126</td>
</tr>
<tr>
<td>Frames and Flexibility</td>
<td>129</td>
</tr>
<tr>
<td>Semantic Networks and Inference</td>
<td>131</td>
</tr>
<tr>
<td>Structured Knowledge</td>
<td>133</td>
</tr>
<tr>
<td>Propositional Networks</td>
<td>136</td>
</tr>
<tr>
<td>A Representational Formalism</td>
<td>138</td>
</tr>
<tr>
<td>Structural Functions</td>
<td>140</td>
</tr>
<tr>
<td>Schematic Function</td>
<td>141</td>
</tr>
<tr>
<td>Value Assignment Functions</td>
<td>141</td>
</tr>
<tr>
<td>Procedural Control</td>
<td>143</td>
</tr>
<tr>
<td>An Example of the Representation Language</td>
<td>144</td>
</tr>
<tr>
<td>Inference Through Association</td>
<td>147</td>
</tr>
<tr>
<td>Goal Condition</td>
<td>148</td>
</tr>
<tr>
<td>Constraint Condition</td>
<td>149</td>
</tr>
<tr>
<td>Decay Rate and Threshold Value</td>
<td>151</td>
</tr>
<tr>
<td>Summary</td>
<td>154</td>
</tr>
</tbody>
</table>

**CHAPTER 6: SPRINT: A MANAGEMENT SUPPORT SYSTEM**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hypermedia Interface for an MSS</td>
<td>157</td>
</tr>
<tr>
<td>A Schematic Memory Metaphor</td>
<td>159</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS (continued)

<table>
<thead>
<tr>
<th>Chapter/Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphical Display and Filtering</td>
<td>159</td>
</tr>
<tr>
<td>Spreading Activation Search Filters</td>
<td>162</td>
</tr>
<tr>
<td>An MSS for Organizational Modeling</td>
<td>163</td>
</tr>
<tr>
<td>An MSS for TQM Strategy</td>
<td>171</td>
</tr>
<tr>
<td>Summary</td>
<td>178</td>
</tr>
</tbody>
</table>

## CHAPTER 7: DISCUSSION

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Achieving the Purposes of a Mental Model</td>
<td>181</td>
</tr>
<tr>
<td>Description</td>
<td>181</td>
</tr>
<tr>
<td>Explanation</td>
<td>183</td>
</tr>
<tr>
<td>Prediction</td>
<td>185</td>
</tr>
<tr>
<td>Representational Capability</td>
<td>187</td>
</tr>
<tr>
<td>Attributing Beliefs to Others</td>
<td>188</td>
</tr>
<tr>
<td>Higher-Order Relations</td>
<td>190</td>
</tr>
<tr>
<td>Evaluating a Model</td>
<td>192</td>
</tr>
<tr>
<td>Modeling Strategic Management Situations</td>
<td>194</td>
</tr>
<tr>
<td>Using SPRINT in Practice</td>
<td>196</td>
</tr>
</tbody>
</table>

## CHAPTER 8: CONTRIBUTIONS AND FUTURE RESEARCH

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Research Contributions</td>
<td>198</td>
</tr>
<tr>
<td>Extending the SPRINT Interface</td>
<td>200</td>
</tr>
<tr>
<td>TABLE OF CONTENTS (continued)</td>
<td>Page</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Mental Models and Coordination</td>
<td>202</td>
</tr>
<tr>
<td>Distributed Knowledge-Based Systems</td>
<td>206</td>
</tr>
<tr>
<td>APPENDIX A: INTERVIEW PROTOCOL</td>
<td>209</td>
</tr>
<tr>
<td>APPENDIX B: NATIONAL QUALITY AWARD CRITERIA</td>
<td>210</td>
</tr>
<tr>
<td>APPENDIX C: DIMENSIONS IN ORGANIZATIONAL MODEL</td>
<td>212</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>213</td>
</tr>
</tbody>
</table>
# LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>The Role of Mental and Computational Models</td>
<td>38</td>
</tr>
<tr>
<td>2.2</td>
<td>Simulating the actual world with a mental model</td>
<td>42</td>
</tr>
<tr>
<td>2.3</td>
<td>Propositional framework: (a) Developing complex mental models; and (b) Strategy formation is dependent on the model structure</td>
<td>53</td>
</tr>
<tr>
<td>3.1</td>
<td>Research Design</td>
<td>67</td>
</tr>
<tr>
<td>3.2</td>
<td>Graphical representation for concept structure</td>
<td>75</td>
</tr>
<tr>
<td>4.1</td>
<td>Representation of concepts in a Hoshin plan</td>
<td>82</td>
</tr>
<tr>
<td>4.2</td>
<td>Concepts and relationships in the TQM strategy</td>
<td>85</td>
</tr>
<tr>
<td>4.3</td>
<td>Concepts and relationships in an organizational model</td>
<td>93</td>
</tr>
<tr>
<td>5.1</td>
<td>A Functional Architecture for Mental Models</td>
<td>102</td>
</tr>
<tr>
<td>5.2</td>
<td>Propositional framework: (a) Developing complex computational models; and (b) Strategy formation is dependent on the model structure</td>
<td>107</td>
</tr>
<tr>
<td>5.3</td>
<td>Knowledge Structure</td>
<td>134</td>
</tr>
<tr>
<td>5.4</td>
<td>Propositional Network</td>
<td>138</td>
</tr>
<tr>
<td>6.1</td>
<td>Fileboxes in an organizational model</td>
<td>165</td>
</tr>
<tr>
<td>6.2</td>
<td>Function hierarchy</td>
<td>166</td>
</tr>
<tr>
<td>6.3</td>
<td>Notes on a function filecard</td>
<td>167</td>
</tr>
<tr>
<td>6.4</td>
<td>Who performs this function?</td>
<td>168</td>
</tr>
</tbody>
</table>
LIST OF ILLUSTRATIONS (continued)

Figure 6.5: What other functions does she perform? .................................. 169
Figure 6.6: Who performs this function? .................................................... 170
Figure 6.7: Fileboxes in the TQM model ...................................................... 173
Figure 6.8: Malcolm Baldrige categories ...................................................... 174
Figure 6.9: Category-Item.Area criteria hierarchy ......................................... 175
Figure 6.10: Complete Item.Area hierarchy ................................................ 176
Figure 6.11: Who is responsible for addressing this Item? ............................. 177
Figure 7.1: Representing the belief of another agent .................................... 189
Figure 8.1: A basic conversation for action .................................................. 207
LIST OF TABLES

Table 2-i: Complementary theories .................................................. 63
Table 5-i: Computational Architecture for Mental Models .................... 120
Table 5-ii: Concept schemata for Organizational mental model ............. 123
Table 5-iii: A frame with its slots and facets .................................... 127
Table 5-iv: A partial Function frame ................................................. 128
Table 5-v: Algorithm for Spreading Activation ...................................... 149
ABSTRACT

The objective of this research is to present a theoretical framework and to describe a computational representation for the mental models that a manager uses when implementing a strategic plan or when attempting to improve an organization's productivity. A mental model consists of some dynamic internal representation that reflects the essential features and relationships in a corresponding real-world system. An integral part of a mental model includes procedures for examining and manipulating its contents and its structure in order to answer questions about the status of goals or the progress of plans. It is suggested that, for an effective manager, these models incorporate a highly integrated network of concepts and propositional relationships that this manager uses to understand strategies and organizations, to infer relationships among indirectly related concepts, and to initiate communication with other managers.

The design for this research includes two case studies which were used to clarify and refine the theoretical concepts about the structure and processing of mental models in strategic management tasks. First, the implementation of a Total Quality Management strategy was studied from the perspective of the Director of Quality Assurance at one division of a large financial services organization. The second study was conducted at a computer manufacturing organization where coordination among functional managers was investigated as they introduced new products to manufacturing.
The computational model developed as part of this research, called SPRINT (Strategic Plan and Resource INTegration), is implemented as a frame-based semantic network using a hypertext interface and is programmed in Smalltalk/V286. This model is used to represent some of the knowledge gathered in the case studies as a means to evaluate the adequacy of the representation scheme and to provide insights into the use of a management support system for similar tasks.

The contribution of this research consists of: (1) clarifying the notion of mental models as used in managerial decision making; (2) specifying a computational representation which accommodates the theoretical framework and the empirical evidence from these case studies; (3) implementing a computational model which embodies the representation scheme; and (4) suggesting how this computational model would be applicable in a management support system for actual management tasks.
What does it mean for a manager to understand the business environment? How does this understanding influence the manager’s ability to explain some phenomenon in the environment or to predict some future state of events? The objectives of this research are to define what this understanding consists of in the strategic management context and, especially, to define a computational means of representing part of this knowledge and its manipulation.

As a brief example, consider a manager faced with the difficult task of downsizing his or her organization in order to reduce costs by 20 percent. A common and often counter-productive method is to cut costs by 20 percent across all departments in the organization. How can an effective, albeit smaller, organization be created? The current strategic objectives must be considered, along with questions such as: Can the current product offering be maintained? Which business functions are most important to the customers and suppliers? Which functions are performed most reliably? How should these attitudes about importance and reliability be considered in recommendations for changing the organization? Are there some functions that should be cut entirely while expanding others? What causes the costs to be incurred? What do customers, suppliers,
and employees feel about current operations? A useful management support system would represent the essential aspects of this dynamic business environment in a form that reflects the manager’s understanding of the organization. In addition, a management support system would include the ability to represent procedural and inferential capabilities that assist a manager in answering these questions.

The objective of this research is to present a theoretical framework and to describe a computational representation for the mental models that a manager uses when implementing a strategic plan or when attempting to improve an organization’s productivity and effectiveness. The research question can be summarized as:

*How can a manager capture and examine, in the form of a computer representation, the complexity of his or her integrated mental model for: (1) a business strategy, and (2) an organization designed to effectively implement that strategy?*

### 1.1 What are Mental Models?

It is often said that, by providing better means for a manager to scan the environment and to filter the constant barrage of incoming information, one can “enhance the executive’s mental model” of the business (Rockart & DeLong, 1988, p. 26). But, what is this “mental model?”

A mental model is not a static set of knowledge but, rather, a dynamic memory that integrates the new information resulting from environmental scanning. Thus, a
mental model consists of some internal representation that reflects the essential features and relationships in a corresponding real-world system. An integral part of a mental model includes procedures for examining and manipulating its contents and its structure in order to answer questions such as those outlined in the above example.

The term 'mental model' has been used with various interpretations in the literature of management, cognitive science, psychology, human-computer interaction, and man-machine systems. As defined in this thesis, mental models consist of mental representations of objects and events in the external world, mental abstractions or concepts, and cognitive processes for manipulating the representations and for conducting mental simulations to explain the world and to predict future events. Simply describing mental representations and cognitive processes does not constitute a new contribution. The contribution of this research lies in: (1) clarifying the notion of mental models as used in managerial decision making, especially for implementing strategy; (2) specifying a computational representation which accommodates the theoretical framework and the empirical evidence from two case studies; (3) implementing a computational model which embodies the representation scheme; and (4) suggesting how this computational model would be applicable in a management support system for actual management tasks. Two well-developed theories form the cornerstones of a framework for these contributions: a cognitive complexity theory for knowledge structure, and an Image Theory for decision making processes.

A complexity-theoretic approach to information processing (Streufert & Streufert, 1978; Streufert & Swezey, 1986) describes the basis for individual differences in
cognitive styles and in cognitive complexity. For example, if two senior executives, both of whom are very effective in one functional area of an organization, are relocated into a different function, one may continue to excel while the other is a notable failure. A theory of cognitive style and complexity describes how the cognitive structure of these two managers may be significantly different, whereas their cognitive content (i.e., task-specific variables) may be similar. This thesis draws on a theoretical description of cognitive structure among flexible, effective managers as a partial specification of the knowledge structure in mental models.

This research is not concerned with the detailed content of the mental models for a particular agent. Rather, it focuses on: the general structure of agent A's mental model for department level strategic plans; A's differentiation and integration of dimensions within his or her planning model; the flexibility with which this integrated network of dimensions is modified to reflect new information; the cognitive procedures that A uses to manipulate the mental model while making strategic decisions; and the degree to which A integrates beliefs of or beliefs about other agents into the model.

The cognitive complexity theory contains few specific proposals for the actual decision making processes used by managers. That is, how is this multi-dimensional, differentiated, integrated knowledge structure used when a manager is confronted with evaluating the progress of a business strategy? Image Theory responds to this question by describing decisions made by individuals acting as decision agents for and within organizations (Beach, 1990; Beach & Mitchell, 1990; Mitchell & Beach, 1990). Images are schemata that represent a decision maker's views in reference to some area of
activity. One image represents basic beliefs and values, another represents goals, and a third represents plans for achieving those goals. The images, by themselves, might be better described by cognitive complexity theory. However, Image Theory contains specific proposals for two kinds of decisions: adoption decisions, which are about adoption or rejection of candidates as constituents of the images; and progress decisions, which are about whether a particular plan is producing satisfactory progress toward its goal. This thesis draws on Image Theory for a partial specification of the decision processes in mental models.

Neither cognitive complexity theory, nor Image Theory, propose a specific mental or computational representation for their respective theories. It was not the intent of those authors to pursue these questions. This thesis synthesizes their descriptions for knowledge structure and decision processes, then proceeds to specify the detail for a computational representation which accommodates their insights. This synthesis forms the basis for mental models used by managers while performing strategic management tasks in an organization. The organization, in turn, is understood as an open system comprised of a loosely linked coalition of relatively autonomous agents influenced by the environment (Scott, 1987). The open system organization provides a context for strategic decisions.
1.2 The Strategic Management Process

The strategic planning domain was selected because it illustrates a classic example of an unstructured decision making task and because it requires multiple agents to coordinate their actions in order to satisfy an organization's goals. This thesis also focuses on department level strategic planning because it illustrates work where interaction among agents is often an ad hoc, on-going, long-term process. It is expected that each individual will have formed mental models in varying degrees of detail about some, but not all, of the other agents involved in implementing the strategic plan. These other agents encompass both individual persons (managers and technical/professional persons) and organized collections of persons (the purchasing department or Acme Corporation). The agents may be either internal to the decision maker's organization or external stakeholders in the plan.

This thesis emphasizes department level strategy (to be distinguished from corporate strategy and business-unit strategy) because it appears to have a degree of specificity and a frequency of change that can provide a useful analysis of mechanisms for coordination. In addition, from an applied research perspective, the department level would probably receive the greatest benefit from a management support system for dynamic strategic planning and decision making. Strategic plans are assumed to exist for each of the agents involved in the coordinated, cooperative effort. These plans are likely to have been negotiated and formalized in face-to-face meetings among relevant groups of managers. The question addressed by this research begins at the point where these
putatively integrated plans are adopted by the respective managers (to whatever level of
detail that the participants feel is necessary). This thesis focuses on the knowledge
structures and decision procedures which support the dynamic adjustment of these plans
in the face of environmental changes, failed commitments, and specification of omitted
details.

1.3 Field Research and Computational Models

The computational representation that is developed to accommodate the
psychological theories must also adequately capture the knowledge and decision processes
used in a real-world system. Field research was conducted in two prominent
organizations in order to clarify my own understanding of a theory for mental models and
to supply data for evaluating the computational model. First, an extended case study was
conducted on the implementation of a Total Quality Management (TQM) strategy by a
financial services organization. The computational model which resulted from this case
study emphasizes the use of criteria specified by the Malcolm Baldrige National Quality
Award as a basis for defining world-class quality. These criteria form part of the
knowledge structure in the mental model used by the Director of Quality Assurance in
this organization. The criteria place thirty percent of their weight on customer
satisfaction, underscoring the importance of the social environment in this strategy. This
study of a TQM strategy was especially interesting due to its timeliness: 180,000 copies
of the document containing the Baldrige Award criteria were distributed by the U.S.
Department of Commerce in 1990, for an award that was only established in 1987. Most of these copies of the criteria document were used to organize an internal assessment of the quality program in each requestor's firm, rather than to write an application for the award. Improving the quality of products and services is currently a top-priority goal for companies nation-wide.

Within this strategic context of TQM, a detailed representation is developed for the generalized mental model that this QA Director has used to assess the effectiveness of more than 70 work groups and organizations. The productivity of an organization may be improved by having people do more of what they should be doing and less of what they should not be doing; that is, more effective at achieving customer requirements and business strategies. The mental model of an organization includes concepts for the functions, people, products, cost drivers, et cetera that support descriptions, explanations, and predictions about the real-world organization. The description of this mental model serves as the basis for evaluating the proposed computational model and management support system.

The second piece of field research consists of structured interviews with twelve functional managers in a computer manufacturing organization. The focus of this work was to document each manager's understanding of his or her role in achieving the overall organization's goals. The particular strategic objectives studied in this case were to reduce the cycle time required for designing new products and for introducing them to manufacturing. The case study was designed to understand the existing strategic planning processes at the department level and to elicit the respective views of each manager about
the interaction among these functional areas. Thus, evidence was obtained for part of the social knowledge in each manager’s mental model and for the planning procedures used to promote coordination among the participants.

This study of coordination assumes that the agents are benevolent; that is, in most cases, each agent is willing to share all relevant information that he or she knows about the current situation. The difficulty arises in the determination of relevance. How does one agent determine which other agents (either organizations or individual persons) are implicated in his current decision-making efforts? What information should be requested of, or communicated to, each agent? How much background information—either general or specific—must be exchanged with another agent to provide the necessary context for successful communication with that agent? This is the aspect of coordination that is addressed in the second case study.

A mental model (or, more likely, a related family of mental models) would be used by a manager to represent and to reason about the knowledge, plans, goals, and values of other agents in order to facilitate coordinated strategy. In addition, the manager would utilize a mental model of the overall organization’s strategic planning process to guide his or her communicative decisions, and a mental model of the organization’s functions, products, cost drivers, et cetera to guide evaluation of current or potential goals and plans. New information gleaned through environmental scanning efforts would be added to one of these dynamic mental models, or used to identify similarities or differences between the model and its real-world counterpart.
This research contributes to recommendations for implementing a management support system to assist managers in strategic management tasks such as those illustrated in the case studies. By embedding the computational representation for mental models into an information system, an "extended memory" may be provided for a manager; that is, a computer-based memory and associated decision processes which are organized along the lines of cognitive complexity theory and Image Theory and which match the knowledge structure used by that manager. It is suggested that departmental managers, or their direct staff members, would use this system for orchestrating the efforts of multiple project teams in the pursuit of a multi-year strategy, such as in implementing a Total Quality Management philosophy in an organization.

The research method employed in this dissertation is designed to evaluate the validity of the computational model, which is based on the mental model concept described above. A case study approach is used to seek evidence for managers' representations of their strategic management environment and for the procedures that they use to examine and manipulate these models. At this early stage in a long-term research stream investigating the nature of mental models for strategic management tasks, it is crucial to retain the contextual complexity of the managers' environments. Thus, field research methods are used which focus on the structure of these mental models. These methods are different from many protocol analyses used in expert systems research that study well-defined decision events. The objective of this research focuses on the strategic management process which spans several days, weeks, or months. Mental
models are knowledge-based resources used in, and modified by, the strategic management process.

In order to illustrate both the representation for mental models and the use of this representation in a management support system, a specific detailed model is developed for assessing an organization's productivity. An organizational model is described in terms of the representation scheme and is implemented in the computational model using data from the case study. The computational model, called SPRINT (Strategic Plan and Resource INTEGRation), is implemented as a frame-based semantic network using a hypertext interface and is programmed in Smalltalk/V286. Conclusions drawn from constructing the computational model for analyzing organizational productivity provide insights into the use of a management support system for similar tasks.

In Chapter 2, a theoretical framework is described which clarifies a mental model's purpose in managerial decision making and which provides a detailed account of cognitive complexity theory and Image Theory within this framework. Chapter 3 describes this thesis' research method as an iterative development and assessment process that specifies mental models from three perspectives: propositions from the theoretical framework, a new set of propositions for the computational representation, and evidence from the case studies. Chapter 4 summarizes the findings from the two case studies and compares the evidence with the theoretical framework in order to assess the validity of the mental model concept. In Chapter 5, a functional architecture is presented for mental models and a four-layer architecture is described for the computational representation. A set of representational propositions are developed which respond to the requirements
laid out in the theoretical framework. The propositions are assessed vis-à-vis the psychological theory and the case study evidence. Chapter 6 describes a prototype system which illustrates how a computational model might be used to implement a management support system for strategic management tasks. Data from the TQM case study is used to exemplify the types of management problems that are amenable to representation in such a system. Chapter 7 discusses the strengths and limitations of the computational representation and the feasibility of a management support system employing this representation. The representation is assessed for its ability to accomplish the three primary purposes of a mental model: description, explanation, and prediction of a corresponding real-world system. Two important aspects of the representational capability are discussed: attributing beliefs to other agents and representing relations associating more than two concepts. Although these capabilities are not tested in the research, a description is provided for how the representation might accommodate these extensions. Finally, Chapter 8 reviews the contributions of this research and outlines plans for future research.
CHAPTER 2

A THEORETICAL FRAMEWORK FOR MENTAL MODELS

Thinking is that facility which

"allows our hypotheses to die in our stead."

-- K. R. Popper

This quote is especially apropos for my use of the term 'mental model'.
Thinking, through the use of a mental model of the world, allows one to reason about possible actions in future scenarios, prior to committing oneself to those actions. Thinking involves the manipulation of symbols that we use to represent the world to ourselves and the operations that we perform on that representation in order to draw some conclusion. This thesis is concerned with the theoretical nature of these representations and operations and with how they are used to define mental models which contribute to strategic decision making.

In order to clarify meanings of the terms 'framework', 'theory', and 'model', the definitions of Anderson (1983) are adopted:
A framework is a general pool of constructs for understanding a domain, but it is not tightly enough organized to constitute a predictive theory. However, it is possible to sample from this pool, tie the constructs together with additional details, and come up with a predictive theory. ... One judges a framework in terms of the success, or fruitfulness, of the theories it generates. ... A theory is a precise deductive system that is more general than a model. ... A model is the application of a theory to a specific phenomenon, for instance, performance of a mental arithmetic task (p. 12).

In this thesis, existing frameworks and theories are employed to define a model for implementing strategic plans. In this chapter, previous uses of the term 'mental model' are reviewed, then a theoretical base is established upon which one can construct a computational representation for mental models.

2.1 The Mental Model Concept

Discussions of “the mental model approach” or “a mental model concept” are pervasive in the literature. Unfortunately, this concept, used by several research disciplines across many domains, has few explicit definitions. Rouse & Morris (1986) state that, “This most likely reflects the extent to which the concept has come to be acceptable on an almost intuitive basis” (p. 349). Intuitive definitions are not sufficient for this exposition on mental models in the management domain; these definitions are especially insufficient for constructing a computational model which reflects the content and processes in a manager’s mental model. The goal of this chapter is to create a definition which can serve as a theoretical specification for a computational representation and model.
The phrase 'mental model' is often used synonymously with 'knowledge'. For example, Porac & Thomas (1990) view mental models as equivalent to "taxonomic cognitive structures." Their defense for this position is that: "Unfortunately, research and theory on managerial cognitive organization is much too undeveloped to clarify the nature of the mental representation of managerial situations. Until more work is completed, we advocate extending a taxonomic analysis of conceptual structure as far as it will go, up to the point where it become inappropriate" (p. 230). I assert that the mental model concept is more than a synonym for knowledge. A mental model, like the theory of which a model is an application, must predict or explain real-world phenomenon. The actual cognitive processes used for this prediction or explanation are likely to be domain specific; thus, a mental model concept must encapsulate the representation of the domain as well as the processes allowing prediction of future states in that domain.

2.1.1 Defining Mental Models

In cases where the mental model concept has been used in an explicit manner, it generally takes on one of two senses. First, mental models are used to represent the domain-specific knowledge needed to understand a dynamic system or natural physical phenomena (deKleer & Brown, 1983; Hayes, 1985). Depending on the model constructed, the knowledge may reflect a naive understanding of the target system, or in-depth expertise about the target's operation. In either case, the model must support qualitative simulation for running the system in the mind's eye.
This first sense may be illustrated in deKleer & Brown's "mechanistic mental model" for a conventional door buzzer. Running the qualitative simulation in their mental model might be expressed as:

The clapper-switch of the buzzer closes, which causes the coil to conduct a current, thereby generating an electromagnetic field which in turn pulls the clapper arm away from the switch contact, thereby opening the switch, which shuts off the magnetic field, allowing the clapper arm to return to its closed position, which then starts the whole process over again. [The repetitive opening and closing of the switch (i.e., its vibration) produces an audible sound.]

Note this simulation's causal nature which qualitatively describes the functioning of the buzzer being modeled. The research in this thesis investigates how such qualitative reasoning might be applied to mental models in the managerial context, such as modeling and simulating the functioning of an organization.

The second sense of the mental model concept is epitomized by Johnson-Laird's work in discourse comprehension (Johnson-Laird, 1983, 1987). A mental model in this sense is used as a mechanism for deductive problem solving by constructing a concrete representation of the premises expressed by a series of assertions and then checking to see what conclusions follow from the model.

Johnson-Laird (1987) describes discourse comprehension as being composed of two stages. First, a superficial linguistic representation of an assertion is constructed using some mental language. Second, this representation, together with general knowledge, is used to construct a mental model of the state of affairs described by the assertion. He criticizes theorists who stop at the first stage; that is, those who use, for example, solely a semantic network for the semantic representation of discourse. A
semantic network, alone, does not describe how words relate to the world. The main result of understanding is that one can imagine how the world would be if the discourse were true. This imaginative process involves a person's ability to manipulate a representation of the world, i.e. a mental model.

I'm interested in mental models of the first sort, i.e. those employing qualitative simulation to describe the functioning of a dynamic real-world system. As will be seen later, these simulations will enable managers to manipulate their models in order better understand the implications of changes in the business environment.

Rouse & Morris (1986) provide the most comprehensive review of mental model literature aligned with the qualitative reasoning sense, although their discussion is decidedly biased toward the manual control research community representing their interests in man-machine systems. Nevertheless, they provide interesting insights into the purposes of mental models. They conclude that mental models must describe, explain, and predict system behavior in any application domain. They propose a functional definition for mental models: "Mental models are the mechanisms whereby humans are able to generate descriptions of system purpose (why a system exists) and form (what a system looks like), explanations of system functioning (how a system operates) and observed system states (what a system is doing), and predictions of future system states" (p. 351). A mental model includes processes that guide retrieval and inference from its representations, or that predict future system states by using procedures learned as part of the mental model.
2.1.2 Managerial Problem Solving

How might these characteristics of mental models—description, explanation, and prediction—be applied to the decision making processes of managers in organizational settings? Both direct and indirect references to the mental model concept in the management domain are reviewed in this section.

As already introduced, Porac & Thomas (1990) have suggested that mental models might provide a useful conceptual tool for understanding the decisions of competitive strategists. Even though the premise of this thesis doesn’t agree with their restriction of mental models to taxonomic structures, they make several useful points concerning managerial cognition. They suggest that decision makers act on a mental model of the environment when seeking explanations for strategic responses to competitive pressures. "Decision makers must have an image of who their rivals are and on what dimensions they will compete" (p. 225). Like this thesis’ emphasis, they focus on the structure of these mental models rather than on the process of competitive identification. However, this thesis goes further by suggesting a more complex structure for representing knowledge and a more complex repertoire of general cognitive processes for conducting mental simulations in these mental models.

Porac & Thomas conclude that, "The structuring effect of a [taxonomic] mental model raises questions about the ability of strategists to reconceptualize competitive environments when patterns of interorganizational relationships change. ... It would seem that one important source of inflexibility is the cognitive inertia that stems from the presence of a well-formed cognitive taxonomy" (p. 237). This idea, that some cognitive
structures encourage inflexible decision making styles, will resurface later in this chapter within the discussion of hierarchical integration in cognitive complexity theory.

The mental model concept was also used to describe the process whereby managers diagnose situations in which employees do not perform up to expectation (Boreham, 1986). In diagnosing the problem, managers generate hypotheses that are subsumed under general organizational categories such as ‘training’ or ‘motivation’. However, ‘these categories were not just labels for groups of hypotheses, but parts of an envisionment or mental model of the organization being dealt with’ (p. 197). They conclude that, ‘The general pattern of the model is an alternation between concept-driven restructuring of a mental model of the situation, and a sensory-driven search through the problem givens. Insights into the structure of the problem obtained by running the mental model during the former type of episode serve as an executive program to direct the search carried out in the latter type of episode’ (p. 205, emphasis added). This thesis also appeals to the notion of running a mental model as a guide to understanding the sensory data in a decision environment.

Mental models may include data triggers which invoke preplanned strategic responses; that is, procedures or plan sequences are activated when a particular situation is encountered in the world-view representation. In order to anticipate future situations, mental simulations might be performed with the mental model in order to observe in the ‘mind’s eye’ which data triggers are activated by a hypothetical scenario. Bourgeois & Eisenhardt (1988) describe how decision execution triggers are used to put structure onto a stream of unstructured decisions in high velocity environments in the
microcomputer industry. For example, a manager might decide to issue new stock, but only if the current stock price reaches some specified value in the future. Such activated procedures provide a link between the mental representation of the environment (real or imagined) and one or more possible action-plans.

Mintzberg (1975) states that a manager has two primary uses for aggregated, “hard” information: building mental models of the things around him, and identifying problems and opportunities. Pounds (1969) describes how a manager might use information to identify the problems to be solved. He defines a “problem” as an association relating the difference between some existing situation and some desired situation. An “operator”, on the other hand, is defined as a managerial activity which transforms a set of input variables into a set of output variables according to some predetermined plan. Thus, since operator selection is triggered by the difference to be reduced, the process of problem finding is the process of defining differences. Problem solving is the complementary process of selecting operators which will reduce differences. The manager defines differences by comparing his perceived reality with the output of a model which predicts the same variables. Therefore, the problem of understanding problem finding is eventually reduced to understanding the models which managers use to define differences.

It is interesting to note that Mintzberg (1987) defines strategic management in a similar vein. The strategist finds strategies no less than creates them by recognizing patterns, often formed inadvertently. Thus, a manager’s mental models of the world may be used for finding problems as well as for finding strategy. A new strategy is often
"crafted" by noticing a new pattern in the competitive environment and, after monitoring it for some time to ascertain its benefit, making the pattern intentional.

The process of strategy formation is one of synthesis rather than analysis (Mintzberg, 1978). This distinction underscores the importance of a mental model representation that supports a holistic, integrated structure through which a manager can recognize the interplay between his organization’s resources and capabilities and the demands or opportunities of the business environment. A mental model records associations resulting from the manager’s "intuition" as well as those from heuristically-based inferences. As described in Chapter 5, a spreading activation search process provides one way for indirect inferences to be drawn from this associative network as a means to explain relationships between the organizational model and environmental dimensions.

The information stored in these networks support the non-programmed activities of upper-management. Executives’ activities are focused on two key processes: agenda setting and network building. Whereas formal plans tend to be explicit, rigorous, and logical, especially with respect to integration of financial goals, a general manager’s agendas often contain loosely connected goals and strategies that address long-, medium-, and short-term responsibilities. A manager builds a network of cooperative relationships with those individuals who play a role in supplying information for developing and implementing his emerging agenda (Kotter, 1982). Each manager possesses beliefs concerning his or her assumptions about the goals, plans, etc. that are held within the mental models of other managers. These assumptions are then tested and modified,
through mental simulation, in an attempt to predict the impact of anticipated actions by these other individuals.

2.2 The Nature of Mental Models

While researchers from many disciplines agree on the value of the mental model concept, there is little detailed descriptive work on what mental models consist of in the management domain. Nor is there sufficient specification for how managerial mental models might be represented, either within the minds of managers or in a computer model. Previous descriptions of the mental model concept have paid insufficient attention to the cognitive processes and procedures which operate on the knowledge structures. This thesis argues that mental models are more than mere representations of the business environment; a mental model (or possibly a family of mental models) is intimately bound with the procedures that operate on it in order to produce descriptions, explanations, or predictions about the environment. The goal is to provide a psychologically based description for this dual nature of mental models and to implement a computational model (i.e., a computer-based model that reflects the structure of a person's mental model) which demonstrates the use of this theoretical description for strategic decision making.
2.2.1 Models in the Mind

Before investigating the specification of a mental model, a more basic question must be asked: What does it mean for someone to have a model in his or her mind? The use of models is inevitable and pervasive. They are found both implicitly in language, economics, and physics, and explicitly in tangible artifacts such as drawings or scaled-down models from architecture and product design. Models are found in thinking itself. In all cases, models assist humans in coping with the unwieldy complexity of the actual world (or expense of bricks and mortar construction in the case of architectural models). If a person were to try and assimilate every detail of the actual world, he or she would simply drown in detail. “Models have to ignore things; that’s how they work. It’s a feature, in others words, not a bug. In fact, that’s what ‘abstraction’ means: it comes from the Latin word for drawing away’” (Davis, 1989, p. 66). Mental models are thus an abstraction and simplification of the actual world.

A computer-based model that attempts to represent objects and systems in the actual world must necessarily be preceded by a mental model of the domain being

![Diagram of mental and computational models](image)

**Figure 2.1: The Role of Mental and Computational Models**

programmed. A human mind must first conceive of, abstract, and simplify the objects,
events, or systems in the actual world before a computer model can be specified and constructed. This dependency is shown in Figure 2.1. At each progressive stage of modeling---actual world to mental model, and mental model to computational model---the objects from the actual world are further simplified and regularized. This simplification is inevitable since computational models cannot, and may never, reach the representational abilities of the human mind. Neuwirth & Kaufer (1989) describe the mental/computational model distinction as internal/external representations. They are concerned with the ability of hypertext-based computer systems (external representations) to augment the performance of persons performing a synthesis writing task by capturing the units and relations (internal representations) from the user's working memory.

The propositions from complexity theory, presented in this chapter as Propositions 2.1--2.16, suggest some guidelines for managers attempting to represent their business environment. These propositions address the center circle in Figure 2.1. The propositions presented in Chapter 5 address the computational model shown as the right-most box in the figure.

A model of the world (either mental or computational) is simplified to include only the objects that the decision maker is interested in, the properties of those objects that he cares about, and the relations between the objects that matter. All other objects, properties, and relations are discarded; that is, ignored or forgotten in the case of a mental model. A model's mental representation specifies how these elements are encoded in human memory, whereas a computational representation specifies how similar elements are encoded in a computer's memory. When a manager models an organization
with the objective of improving its productivity, he or she does not represent the height of the employees, the color of their eyes, or the personal incomes of its customers. Models also reflect individual differences in decision making style or responsibility. A model conceptualized by one manager may only consider objects that can be quantified in financial terms, whereas another manager might limit himself to concerns about organizational ecology, affirmative action, and employee empowerment. Each model leaves out concepts that are important in some identifiable context other than the one for which it was intended. Thus, consideration of a given model's 'adequacy' or 'completeness' is pointless unless the context of its application is also specified.

"In contrast to modeling, something that's not partial is action. Actions are not constrained to operate within the particular level of abstraction at which the model is formulated. Actions are full-blooded" (Davis, 1989, p. 70). You don't act in a model, although you may construct a model of an action: (1) before the act is committed in order to predict the desirability of its anticipated outcome, or (2) after the act in order to explain why it occurred. These models of action are considered under the category mental simulation.

The following two sections provide further insight into the two principle components of mental models: mental representation and mental simulation.

2.2.2 Mental Representation

The fundamental problem of representation is: How do states of the mind encode corresponding states of the world? Or, if one subscribes to an information processing
or computational theory of cognition: How can these mental representations be encoded in a corresponding computer system? Those researchers who are interested in the foundations of artificial intelligence might consider theories spawned from the former question as motivation for answers to the latter. This thesis is firmly grounded in this foundationalist approach to artificial intelligence.

At the most basic level, mental representation might be grouped into two potentially competing categories: classical symbolic representation, and connectionist representation (Fodor & Pylyshyn, 1988). This thesis doesn’t take a stand on one position or the other, but chooses the higher-level, more well-developed symbolic position in order to achieve the pragmatic concern for developing a management support system.

Among symbolic representation schemes, there are a host of alternative paradigms for capturing an internal representation of real-world objects, events, or systems. A discussion of this choice is deferred until Chapter 5, after the theoretical foundation for mental models has been completed and the empirical evidence from the case studies has been presented. Additional literature related to representation is reviewed at that time.

2.2.3 Mental Simulation

Most proposals for knowledge-based systems are evaluative in nature. The systems usually represent process-oriented knowledge, often in the form of production rules, that is used to recommend some action when given data about the current state of the world. What if these recommended actions are not performed in the actual world
but, rather, in a mental representation of the world, i.e. a mental model? As the introductory quote from Popper suggested, thinking is that facility which "allows our hypotheses to die in our stead"---by testing alternative actions through mentally simulated actions on our mental model of the world. Such is the case for managers who try out alternative strategies, mentally, prior to committing their organizations' resources.

A person's mental architecture may consist of two modules: a set of evaluative or interpretive procedures, and a model of the world (Rumelhart, Smolensky, McClelland & Hinton, 1986; Rumelhart, 1989). Figure 2.2 illustrates the cooperative nature of these two modules. The outputs from the evaluative module, i.e. recommended actions, are redirected to the model of the world which predicts their effect in the actual world. This
prediction then serves as a new input to the evaluative module which may recommend further actions. Thus, we “run a mental simulation” by cycling between these two modules until a satisfactory state of the world is obtained within the mental model.

Several theories have been described which suggest the use of mental simulation. Hogarth & Einhorn (1990) develop a descriptive theory that accounts for choice behavior in situations involving risk and uncertainty. The key proposal of their “venture theory” is that:

Decision weights used to discount values of outcomes for uncertainty are the end result of a process that involves first anchoring on an estimate of probability and then adjusting this by imagining other possible values for the weights. ... This adjustment is the net effect of a mental simulation process in which the decision maker “tries out” various weights suggested by different possible scenarios. ... It seems reasonable to assume that the greater the payoffs, the more people invest in mental simulation, i.e., they think more about large as opposed to small payoffs (p. 782).

It has also been suggested that mental simulation may account for the process by which one individual interprets the mental states of, or ascribes mental states to, other persons (Goldman, 1989). Interpreters might ascribe mental states to others by “pretending or imagining themselves to be in the others’ shoes, constructing or generating the (further) state that they would then be in, and ascribing that state to the other. In short, we simulate the situation of others, and interpret them accordingly” (p. 169). In closer alignment with the proposal of Rumelhart, et al. described above, Goldman also suggests that, “Simulation is also relevant in inferring actions from mental states, not just mental states from other mental states” (p. 170). In this case, the
"model of the world" shown in Figure 2.2 would correspond to a model of another person.

This thesis takes one small step toward describing a computational representation for a manager's mental model of an organization, including simulating the outcomes of actions taken in those models. I emphasize the structure of a mental model, and describe some processes for manipulating a mental model, but make no commitment to executive control processes for automatically "running" a simulation in a mental model. At this point, my attention is restricted to what might be considered one iteration of a mental simulation cycle, i.e. making a change in a mental model of an organization and observing the effect of that action.

2.3 Organizational Psychology

Now that a framework has been laid for the role that mental models play in managerial decisions and for the high-level structure of mental models in terms of mental representation and mental simulation, the structure and processes of these mental models must be analyzed in greater detail. Toward this end, two well-developed theories are used as the cornerstones of a framework for representing mental models. First, cognitive complexity theory contributes to understanding the knowledge structure used by effective decision makers. Second, a particular form of decision theory, called Image Theory, is presented as being descriptive of decision processes used in individuated decision events.
One important common emphasis shared by cognitive complexity theory and Image Theory is that both describe individual decisions made by individual decision makers, not aggregates of decisions or aggregates of decision makers. This emphasis is critical for appreciating a primary purpose of this thesis: To represent a psychologically plausible model for the knowledge structures and processes used in specific decision instances by individual managers. However, as described more fully in Chapter 5, the psychological plausibility is intentionally curtailed in some cases in order to achieve the dual purpose of a realizable management support system using current information technology.

As further emphasis for this point, Beach & Mitchell (1990) carefully distinguish traditional decision theory, e.g. Expected Utility Theory, from their own proposed Image Theory.

The [traditional decision] theory's limited descriptive sufficiency results from the fact that it is more useful at some conceptual levels than at others. Thus, as a broad heuristic it seems to be fairly useful for conceptualizing aggregate decision behavior. On the whole, creatures tend to act in their own best interests and traditional theory therefore is useful for rationalizing the data of foraging animals or the behavior of economies. That is, for aggregates of single decisions by a single individual, traditional theory often is descriptively sufficient. ... At more concrete (molecular) conceptual levels, for example at the level of individual decisions and of decision makers' self report about how and why they make the decision they did, traditional decision theory proves to be descriptively insufficient. (pp. 2-3)

A similar point is made by Streufert & Streufert (1978) in defense of their approach for studying the varying abilities and styles that individuals use in coping with
the complexities of the real world. This same concern must be central to the study of managers coping with the complexities and equivocalities of strategic management.

We should recognize that individuals have different ways of dealing with multidimensional environments and their complexities. We must be aware of the possibility that some of the variance in our data may be created by these "individual" differences. What needs to be measured may, in many cases, be the effects of the interaction between the complexity of the environment and the coping of the subject. (p. viii)

If we intend to work with multidimensionality, we have to permit our subjects to determine their individual dimensionality and let them work with that, rather than let other or average subjects' responses determine the dimensions. (p. 6)

With these concerns in mind, the two theories are presented in the following two sections. Greater attention is devoted to the cognitive complexity theory since this thesis places primary emphasis on the structure of mental models rather than on specific decision procedures. These descriptive theories are used as a prescription for the representation of mental models which can accommodate the psychological processes actually used by managers.

2.3.1 Cognitive Complexity Theory

Cognitive complexity theory doesn't describe a particular theory as much as a related family of theories which address individual differences in cognitive structure. Streufert & Streufert (1978) have attempted to clarify this research field by resolving discrepancies in terminology.

While several of the complexity theories contain unique aspects, a core of common orientations exists as well. Complexity theorists and researchers have been concerned with the dimensionality of human cognition. ... All
would agree that a consistent, i.e. unchanging, response despite contrary information would suggest that information has been processed in unidimensional fashion. For example, if 'John' were always categorized as a 'bad' person, regardless of context, task or situation, unidimensional (undifferentiated) cognition would be implied. ... Hierarchic integration would suggest a complex but relatively fixed and consistent mode of processing information. Most would emphasize 'flexible complexity', the capacity to reintegrate with change in relevant information as the primary form of more advanced human information processing (Streufert & Nogami, 1989, pp. 106-107).

Cognitive complexity is not concerned with 'what' people think; rather, it focuses on 'how' people think (Streufert & Nogami, 1989). For example, consider two managers who are equally effective decision makers in one organization. If they are both transferred to a new organization, one may continue to excel while the other is a notable failure. In spite of the fact that these two individuals have roughly equivalent content in their respective knowledge of the organizational environment, the structure of their knowledge is importantly different.

The distinction between structure and content is critically important to complexity theory. The theory focuses on the structure of information processing; on how information is processed. Cognitive content is concerned with the location of stimulus objects on a given dimension rather than with the structural relations among dimensions. While both cognitive structure and attitude content are important when studying decision outcome, complexity theorists emphasize the former over the latter.

A complexity theoretic approach to information processing (Streufert & Streufert, 1978; Streufert & Swezey, 1986; Streufert & Nogami, 1989) provides useful terminology for describing and assessing the knowledge structure of an individual’s mental model.
A *dimension* in a mental model is a bipolar cognitive scale with two or more points of discrimination among stimuli, i.e. shades of grey, which represents the grouping or ordering of cognitive concepts. Examples of dimensions would include "profit" and "productivity." A minimal dimension has just the two end-points, which enables inclusion-exclusion principles in perception and reasoning. *Discrimination* is the process of dividing a dimension into subsections and is meaningful only to the degree that sharp distinctions can be made when placing stimuli on this cognitive scale. *Differentiation* is the process of dividing a cognitive space (i.e., the representations in a mental model) into two or more orthogonal bipolar dimensions. A differentiated knowledge structure provides in some sense first-order complexity; the decision maker chooses one dimension from among several alternatives.

The most crucial aspect of complexity is that a mental model is *integrated* to the extent that two or more differentiated dimensions are related to produce an outcome that is determined by the joint (weighted or unweighted) demands of each dimension. Integration can take two forms: hierarchical and flexible. In hierarchical integration, specific stimuli would always affect the same dimensions in the same way and those dimensions would be combined in a fixed, consistent manner. Flexible integration is responsive to anticipated changes in the environment that require reconceptualizations of the concept relationships. However, flexible integration is not always preferred over hierarchical styles. Fixed input with given meanings that can be quantified in advance are best accommodated by hierarchical complexity, while novel, dynamic environments
that include unexpected components are most compatible with flexible complexity. Strategic management tasks are generally of the latter variety.

The role of integration in strategy development is summarized by Streufert & Swezey (1986, p. 75):

The development of strategy is not an emotional or irrational process. It is rational, but not in the terms of narrow mathematical definitions. Mathematical models, so popular in decision theory, reflect a hierarchical integrative approach—an approach that has difficulty dealing with change and is incapable of dealing with unforeseen events. Yet, unforeseen events must be dealt with frequently by organizations and managers as the task environment changes more and more rapidly with time.

Cognitive complexity—simplicity represents the degree to which a potentially multidimensional mental model is differentiated and integrated. A cognitively complex person’s mental model would likely function multidimensionally. A less complex person would respond to stimulus arrays on the basis of few or only one dimension; that is, the person would display little or no dimensional differentiation and integration. However, complexity in decision making is not a case where more is always better. There are situations where complexity is not warranted; for example, in crises where time is more important than comprehensiveness, or in situations of cognitive overload (Streufert & Swezey, 1986).

An intuitive understanding of cognitive complexity may be enhanced by considering a culinary example. According to Ceci (1990, p. 23), ‘The construct ‘elaborated knowledge domain’ refers to a knowledge structure that is organized by a relatively large number of dimensions. These dimensions are integrated in such a fashion that fine-grained classifications can be made, as in the case of someone who can
differentiate a variety of foods (dairy, meat, dessert, etc.) along different dimensions, and the dimensions themselves can be integrated to produce various combinations (full course dinners, Indian foods, etc.). ... A complexly represented object is one that is assigned to many attributes or dimensions which themselves can be temporarily associated (i.e., integrated) to make sense of novel data or to distinguish 'shades of gray'.”

Current information systems cannot adequately represent flexible integration as described by cognitive complexity theory. Database systems provide a repository for the dimensions of a problem domain, but differentiation and integration are only available through externally defined queries. The integrative features of the domain are not embedded within the model itself; the integrative expertise is supplied by the user when querying the database. The knowledge representation and the prototype system described in this thesis provide a way for an individual to add new dimensions to a model, integrate the structure, and embed heuristic procedures within a single computational model that corresponds to the user’s mental model of a real-world system. A set of processes are described which permit flexible manipulation and reorganization of the model’s structure.

A descriptively accurate knowledge representation would capture the cognitive complexity of an individual’s mental model, whereas a normative approach would prescribe how an individual’s model should be multidimensional, differentiated, and integrated. The primary emphasis in this thesis is to describe a representation for an individual’s actual model of the world, but some recommendations are provided for building a system which suggests how an individual’s models might be more highly differentiated and integrated, or how an individual’s model might be evaluated for its
degree of differentiation and integration. As pointed out by Streufert & Swezey, the presence of perceptual differentiation and integration in an individual’s mental model does not necessarily imply that differentiated and integrated decision-making activities will follow. An information system may fill a prescriptive role in this situation since a system that represents a complex model may also encourage the person to consider this complexity when making decisions.

2.3.2 Complexity-Theoretic Propositional Framework

The propositions for relating cognitive complexity theory to mental models are loosely organized into a process-oriented framework describing the formation and use of mental models in strategic decision making. As shown in Figure 2.3(a), a new dimension is often created simply to include or exclude some environmental cue. Then, the dimension may be elaborated by discriminating values within its bi-polar scale. Over time, the decision maker differentiates by learning of alternate dimensions that may be independently employed in the decision task. However, if too many alternative dimensions are considered, the decision maker might suffer from cognitive overload and fail to develop integrated strategies; excessive differentiation side-tracks the model formation process.

Within a reasonable degree of differentiation, the model progresses toward integrating environmental stimuli into low- and high-level concepts. Further development depends on whether the model structure enables hierarchical or flexible knowledge structures. A hierarchical structure may be best for, but limited to, steady-state decision
environments, whereas flexible integration supports further model development toward decision contexts that can be reopened for future examination and modification. Such flexibly integrated structures, and the original creation of new dimensions, depend on the mental model being open to additions and reorganizations in its structure.

Figure 2.3(b) illustrates how the strategy-formation process is dependent on the general process for developing cognitively complex mental models. A variety of cognitive styles may be used by a manager when forming strategy, minimally employing a single dimension, but really beginning with differentiated styles. However, according to complexity theory, an "integrative mandate" asserts that true strategy development is not possible without integration.

In order to ease information overload, the strategist forms both low- and high-level composites of some selected dimensions in order to focus attention on higher-level integrations when developing a strategy. A flexible strategy should maintain several contingent alternatives "on hold" while remaining sensitive to continuous information flow in the decision environment. The forthcoming discussion of Image Theory directly addresses this need for evaluating the status and progress of existing goals and plans.

The principal propositions---those most relevant to the research in this thesis---from cognitive complexity theory are introduced in the next two sections. These propositions are taken verbatim from Streufert & Swezey (1986). Each proposition ends with a parenthetical reference to the number of that proposition in the original text.
2.3.2.1 Developing complex mental representations

**Proposition 2.1 (INCLUSION):** Early experience typically involves cognitive conceptualizations that employ inclusion-exclusion principles. (S&S 4.8)

**Proposition 2.2 (DISCRIMINATION):** Subsequent experience and/or training can lead to the development of unidimensional conceptualizations.
that include discrimination. These conceptualizations are typically fixed with regard to specific sets of stimuli. (S&S 4.9)

**Proposition 2.3 (DIFFERENTIATION):** Experience with alternate conceptualizations of stimulus sets may lead to development of alternate dimensions of judgement (differentiation), which may be employed independently, cued by specific stimulus or cognitive conditions. (S&S 4.10)

**Proposition 2.4 (EXCESSIVE DIFFERENTIATION):** Excessive use of dimensionality (excessive differentiation) may generate confusion concerning which dimensions to apply to specific stimuli or cognitive events and may prevent development toward integration. (S&S 4.11)

**Proposition 2.5 (LOW-LEVEL INTEGRATION):** Experience or training with possible relationships among differentiated conceptualizations of stimuli may lead to low levels of integration where diverse cognitions are related and combined into an overall view (e.g., a strategy). (S&S 4.12)

**Proposition 2.6 (HIGH-LEVEL INTEGRATION):** Experience or training with interrelationships among various integrated conceptualizations or strategies may lead to higher levels of integration where several (low-level) integrated conceptualizations are combined into metaconcepts, long-term goals, and so forth. (S&S 4.13)

**Proposition 2.7 (STRUCTURAL):** Both the number of dimensional concepts (differentiation) and their relationships (integration) can be either fixed and hierarchically organized, or can be flexible and open to modification with additional informational input. (S&S 4.5)

**Proposition 2.8 (HIERARCHICAL LEARNING):** Hierarchical complexity is emphasized and likely generated when an entire system of dimensions and relationships is presented to an individual at one time, and/or when the system is presented as invariable. (S&S 4.6)

**Proposition 2.9 (STEADY-STATE):** Where differentiation and integration are hierarchical (organized without flexibility in response to subsequent information, see 2.2, 2.3, and 2.4), a person will function appropriately only in a steady-state environment. (S&S 4.14)

**Proposition 2.10 (FLEXIBLE LEARNING):** Flexible complexity is emphasized and likely generated by encouraging an exploration of the components and relationships existing within a system, by permitting
developmental explorations, and by de-emphasizing memorization for the system. (S&S 4.7)

Proposition 2.11 (REOPENABLE CONTEXTS): Decision makers who employ high levels of integration are likely to be effective only if they are able to close on decisions when required, despite remaining uncertainty. Effective integrators are likely to reopen their cognitive considerations after a decision has been made (where a previous decision may yet be modified) and can (where useful) make adjustments or other modifications to the previous decision. (S&S 4.15)

Proposition 2.12 (OPENNESS): Development of cognitive complexity in domains where complexity does not presently exist can be aided by discovery of multidimensionality in the environment, as long as (1) the person involved is open to the potential existence of additional dimensions, and (2) sufficient flexibility to permit reorganization of relevant cognitive concepts is present. (S&S 4.4)

2.3.2.2 Complexity and strategy formation

Proposition 2.13 (COGNITIVE STYLE): Cognitively less complex managers who function in structurally unidimensional fashion may conceive of strategy as the fixed application of a given principle to a specific problem. Managers who employ structurally differentiated styles typically employ a pragmatic (one-at-a-time solutions) approach. Managers who are flexible integrators tend to plan across several steps toward a goal, keeping alternative steps in mind as contingencies. (S&S 5.19)

Proposition 2.14 (INTEGRATIVE MANDATE): True strategy development is not possible without integration. (S&S 5.22)

Proposition 2.15 (COMPOSITE FORMATION): Strategy development requires consideration and reconsideration of current and potential events, conditions, and outcomes. To avoid the information overload that may be associated with this process, integrative efforts that transform these events, et cetera, into composites will effectively lighten the load, allowing strategy building via higher-level integrations of previously integrated concepts. (S&S 5.25)
Proposition 2.16 (FLEXIBLE STRATEGY): Strategy development may be of little benefit (1) if it is inflexible, (2) where, based on current conditions, it extends over too many steps, (3) where too few contingent and flexible alternative actions under conditions of excessive uncertainty are considered, or (4) where strategy development occurs in a vacuum—that is, remaining apart from sensitivity to continuous information flow. (S&S 5.28)

A mental model does not excel through its knowledge structure alone. A set of processes must exist for manipulating the model, for storing and retrieving both content and structure-related information, and for performing mental simulations. These processes must ultimately achieve a mental model’s purposes, i.e. to describe, explain, and predict the decision maker’s environment. “Cognitive processes and elaborated knowledge domains are best viewed symbiotically: Efficient cognitive processes help add structure and complexity to an existing knowledge domain, and in turn, this structure may enhance the efficacy of the cognitive processes that operate on it.” (Ceci, 1990, p. 24). A discussion of these processes is deferred until Chapter 5.

For another view on how mental models might actually be used in decision making, I turn next to a complementary theory which describes strategic decision making processes.

2.3.3 Image Theory

As introduced earlier, a decision theory that is appropriate for representing mental models should describe the cognitive processing of individual managers when making single decisions; the theory should account for individual differences among managers.
Image Theory fulfills this requirement and provides a theoretical basis for mental models that is complementary to cognitive complexity theory.

The basic premise behind Image Theory is that decision making managers act as promoters and protectors of the organization's values rather than as relentless seekers of maximal payoffs. Managers tend to resist change for its own sake and to maintain the status quo as long as it is compatible with the organization's goals, plans, and values (Beach & Mitchell, 1990; Mitchell & Beach, 1990; Beach, 1990; Mitchell, Rediker & Beach, 1986; Beach, Smith, Lundell & Mitchell, 1988). Strategy formation can be viewed as the interplay between a dynamic environment and a bureaucratic momentum, where the role of leadership is to mediate between these two forces by maintaining the stability of the organization while at the same time insuring its adaptation to environmental change (Mintzberg, 1978). Image Theory was developed as an alternative to traditional decision theory due to the traditional theory's failure in accounting for the ways that people actually make most decisions.

2.3.3.1 Images

The primary structural concepts in the theory, images, are schemata that represent the decision maker's personal values and his knowledge about the organization and its strategies.

The value image describes both the personal values of the decision maker (ethics, morals, etc.) and the decision maker's understanding of the organization's principles (e.g., serve the customer, maintain profitability, maximize market share). The value
image is analogous to the organization's culture representing the members' shared principles pertaining to the organization. The value image guides the decision maker in adopting constituents in the other two images.

The trajectory image describes the decision maker's strategic vision and consists of the organization's goals, both concrete (achieving a given profit) and abstract (gaining a reputation as the industry leader).

The strategic image describes the plans that the decision maker has adopted for attaining the goals in the trajectory image. This image also includes forecasts about the plans' expected outcomes. In following Proposition 2.13 from cognitive complexity theory (COGNITIVE STYLE), the strategic image is considered to include both active plans and inactive, contingent plans.

Image Theory provides no further detail about the structure of individual images, nor about how these images are related to one another. For this purpose, I appeal partially to cognitive complexity theory and partially to the representational propositions developed in Chapter 5. Complexity theory describes how the dimensions within these images should be differentiated and flexibly integrated for high-performing managers. The representational propositions suggest how the image schemata might be internally structured and associated with one another.

2.3.3.2 Decision framing

When a manager is confronted with a decision, he or she draws on prior decision contexts that were identical or similar to the present circumstances, and/or chooses
relevant images that pertain to the current situation. A meaningful decision context is referred to as a *frame* within Image Theory.

Perceptual cues from the decision environment are used by the decision maker to search his or her memory of past experiences and images in order to create a frame for the current decision. If a prior context (consisting of images used in a previous decision) is located which is identical to the current environmental attributes, then the context is said to be *recognized*. If, on the other hand, one or more prior contexts are located which only partially resemble the current context, then those contexts have been *identified*. Both types of memory retrieval permit the decision maker to relate the current environment to memory. However, a recognized context, if it were previously successful, may permit the decision maker to automatically implement the goals and plans of that prior context, while an identified context only serves as a basis for the current decision.

2.3.3.3 Three types of decisions

Once a decision frame is formed for the current environment, its images serve as inputs to three kinds of decisions.

*Status decisions* are goal-centered in that they evaluate the compatibility between a particular goal and its corresponding measurement in the current decision environment; that is, deciding whether the organization is achieving its goals. A goal in the trajectory image might be specified to correspond with one of an organization’s performance measures. The performance measures monitor an organization’s critical attributes
without necessarily specifying a corresponding plan in the strategic image. A plan may be adopted only in the event that the goal is not compatible with the environment. (Image Theory only proposes two types of decisions, omitting status decisions. I have added this decision type based on data collected in the Total Quality Management case study, described in Chapter 4.)

*Progress decisions* are plan-centered in that they evaluate whether a particular plan in the strategic image is forecasted to produce satisfactory progress toward achieving its associated goal in the trajectory image. Unsatisfactory progress may prompt a modification of the goal or a change in the plan in order to rectify the imbalance.

*Adoption decisions* accept or reject candidates as constituents of the value, trajectory, or strategic images.

### 2.3.3.4 Two decision tests

Two decision tests are proposed by Image Theory to support the three types of decisions.

The *compatibility test* is always the first step in any of the decisions. In adoption decisions, the candidate is assessed for its “fit” with the decision maker’s three images. It fits if the disparity between its attributes and the images does not exceed a rejection threshold value. If only one candidate passes the compatibility test, it is adopted without further analysis. In cases where more than one candidate pass the first step, then the second step, the *profitability test* is invoked to select one of the candidates. It is assumed that the decision maker has a repertory of strategies for choosing the best candidate from
among a set of two or more candidates, all of which are minimally acceptable. The profitability test is considered to be more deliberative, and usually more complex, than the compatibility test. The profitability test is only invoked for adoption decisions.

Status decisions evaluate the “fit” between the trajectory image and the decision environment. Progress decisions determine whether the strategic image’s forecasts are sufficiently compatible with the trajectory image’s goals to warrant continuation of existing plans.

In all cases, the compatibility test determines “fit” by tallying the number of violations, where each violation is all-or-none. A violation occurs when some attribute of a candidate or image negates, contradicts, contravenes, prevents, retards, or in some other way is contrary to or interferes with an image constituent. The decision rule is that if the number of violations exceeds some absolute rejection threshold, the fit is rejected.

Note that the default value for the compatibility test is “yes.” Unless violations exceed the rejection threshold, candidates are eligible for adoption or progress is assumed satisfactory. The compatibility test favors the maintenance of the status quo and works against abrupt changes in the organization’s direction and activities (Beach & Mitchell, 1990).

In summary, Image Theory describes a part of the knowledge that must be represented with a mental model (i.e. the three images), several manipulations that must be performed within mental models (i.e., the decision framing process), and decision heuristics for using the mental model in strategic decision making (i.e., the three kinds of decisions and two decision tests). This thesis emphasizes the knowledge structures and
general manipulations of mental models over the details of particular decision heuristics. Thus, the decision procedures will not be specified as part of the computational representation presented in Chapter 5.

2.3.4 Complementary theories

The complementary nature of cognitive complexity theory and Image Theory is summarized in Table 2-i. The items that are horizontally adjacent to one another indicate rough analogues in the respective theories. By combining the contributions of these two theories, I hope to provide a complete framework for the structure and use of mental models in managerial tasks.

2.4 Summary

This thesis reports a multimethod research project which surveys the extant literature for a theoretical framework that describes mental models in managerial decision making and which uses two case studies involving three situations to build and evaluate a computational model. The cognitive complexity theory and Image Theory were chosen for their descriptive nature and for their specificity about the cognitive aspects of strategic management. The theoretical framework described in this chapter serves as a prescription for specifying a computational representation and for designing a computational model. The primary outcome of this work is a computational representation for building computational models that reflect the mental models actually
<table>
<thead>
<tr>
<th>Cognitive Complexity Theory</th>
<th>Image Theory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use multiple cognitive dimensions</td>
<td></td>
</tr>
<tr>
<td>Discriminate values within dimensional scales</td>
<td></td>
</tr>
<tr>
<td>Differentiate among dimensions</td>
<td></td>
</tr>
<tr>
<td>Integrate dimensions hierarchically</td>
<td></td>
</tr>
<tr>
<td>Integrate dimensions flexibly</td>
<td></td>
</tr>
<tr>
<td>Perceive stimulus configuration</td>
<td>Frame context and events of the decision</td>
</tr>
<tr>
<td>• Observe passively</td>
<td>• Relevant value image principles</td>
</tr>
<tr>
<td>• Search actively</td>
<td>• Relevant trajectory image goals</td>
</tr>
<tr>
<td>Differentiate and integrate attributes of stimulus objects with cognitive dimensions</td>
<td>• Relevant strategic image plans</td>
</tr>
<tr>
<td>(perceptual-information-acquisition functions)</td>
<td></td>
</tr>
<tr>
<td>Differentiate and integrate cognitive dimensions when making decisions</td>
<td>Evaluate status of goals</td>
</tr>
<tr>
<td>(decision-making-executive functions)</td>
<td>• Assess compatibility</td>
</tr>
<tr>
<td></td>
<td>Evaluate progress of plans</td>
</tr>
<tr>
<td></td>
<td>• Forecast outcome</td>
</tr>
<tr>
<td></td>
<td>• Assess compatibility</td>
</tr>
<tr>
<td></td>
<td>Evaluate adoption</td>
</tr>
<tr>
<td></td>
<td>• Recognize and implement policies from prior decision frames</td>
</tr>
<tr>
<td></td>
<td>• Identify and reformulate policies from similar decision frames</td>
</tr>
<tr>
<td></td>
<td>• Assess compatibility</td>
</tr>
<tr>
<td></td>
<td>• Assess profitability</td>
</tr>
<tr>
<td></td>
<td>• Generate new candidate principles, goals, and plans</td>
</tr>
<tr>
<td></td>
<td>• Assess compatibility</td>
</tr>
<tr>
<td></td>
<td>• Assess profitability</td>
</tr>
</tbody>
</table>

Table 2-i: Complementary theories
used by managers.

This chapter was not intended to survey all literature referred to in this thesis; only the literature applicable to the theoretical framework is reported here. Chapter 5 is devoted to all aspects of representing mental models in a computer-based model. Further literature is reviewed there as it relates to frame and semantic network representations. A representational formalism and a spreading activation search algorithm are developed in that chapter.
CHAPTER 3

ESTABLISHING THE NATURE OF MENTAL MODELS

The previous chapter defined the mental model concept for managerial decision making. However, it still remains to be shown that this is a valid concept and that a representation for mental models is useful when designing computerized management support systems. In order to determine the validity of the mental model concept, the role that these models play in managerial decision making must first be established. Can a manager's knowledge be encapsulated within one or another of these mental models, or is the knowledge inseparably interwoven into one web of general domain knowledge? Where does one draw the conceptual lines around a mental model (that is, where does one model begin and another leave off)? Are they mutually exclusive or overlapping? How do these models interact with one another during the decision making process? The notion of encapsulation that I use here is different from modularity in basic cognitive processes, such as the separation of language processing, visual processing, etc. (Fodor, 1983). I am interested in the encapsulation of domain knowledge into separate mental models that correspond to real-world systems considered when making decisions.

It might be hypothesized that the mind is somehow divided into several models which represent their respective objects or social systems in the real-world. For
example, separate models might exist for: the work flow within a decision-maker's organization structure; other agents' beliefs and preferences (e.g., competitors in the marketplace); or the concepts and organizational procedures used to develop and implement a strategic plan. Each of these mental models may be, at least partly, distinguishable and autonomous from the others.

If a mental model is discernable from general domain knowledge used in a manager's day-to-day decision making tasks, one should be able to observe subdomains that the decision maker, S, treats in a relatively independent manner. That is, a regular pattern might be observed for the way that S decomposes decisions into several subdomains, manipulates a model of each subdomain, and then integrates these partial solutions into the final decision. These subdomain models should correspond to identifiable real-world systems, and each mental model should serve as a surrogate for the real-world system in the manager's decision-making process.

Since I am interested in the interactive complexity of relatively large subdomains (such as S's understanding of the strategic planning process, or S's understanding of an organization's capacity to fulfill its strategies) and in the integration of these subdomains into the complete decision process, my research objective can be best served by studying S's decisions in their natural context. Two case studies were designed to investigate the nature of mental models and to establish the validity of this concept.
3.1 Research Design

The research design used in this dissertation is summarized in Figure 3.1. The primary triad in this design consists of an iterative development cycle among three areas:

(1) developing a theoretical framework for mental models, (2) conducting case study research, and (3) specifying a computational representation for mental models. Initial theoretical concepts were reviewed in the last chapter which suggested the potential benefit for constructing a representation of mental models. These concepts were drawn from management literature (Porac & Thomas, 1990; Pounds, 1969; Kotter, 1982; Mintzberg, 1975, 1978, 1987) that described the general characteristics of managers who formed mental models of their environment (although not all the authors explicitly used the term ‘mental model’).
Two cases were selected that would illuminate the theoretical propositions. While the case studies were progressing, I began to specify a computational representation which addressed the theoretical framework for mental models. Evidence collected in the case studies served to refine both the framework and the representation, and questions arising in the representation provoked additional investigation in the case studies and in the framework. The case study research design is described in the next section.

The theoretical propositions presented in the last chapter serve as the "backbone" for both summarizing the case study evidence in Chapter 4 and for developing corresponding propositions about the representational constructs presented in Chapter 5. This common theoretical backbone provides a basis for assessing the similarity and the differences among these three foundations for the primary triad in this research design.

Using evidence from the case studies, and the specification for a computational representation, a computational model was implemented in the Smalltalk/V programming environment. Just as the case studies have assisted me in refining the theoretical framework's details, implementing a computer-based model of my proposed computational representation forced me to clarify and refine the representation. Programming a computer model is a method in this research design rather than an output of the research. A similar motivation has been explicitly identified by many notable researchers in cognitive science (Anderson, 1983; Johnson-Laird, 1983; Pollock, 1987; Thagard, 1988).
In describing why knowledge-based systems are a good medium for dealing with incompletely understood tasks and reasoning, Davis (1989, p. 73) summarizes a similar motivation within the artificial intelligence research community:

The programs we write will probably not be the ultimately important thing, though extensive pragmatic benefit may result from them. Perhaps more important will be the role of system construction in encouraging the accumulation, organization, systematization, and further development of knowledge about a task.

Davis’ description is appealing and particularly relevant to research on information systems support for unstructured tasks such as strategic management. By developing a systematic means for organizing a manager’s knowledge—in terms of a mental model’s knowledge structures and cognitive processes—one can accumulate a consistent body of data about these tasks in many different contexts. Any particular programmed artifact will quickly become obsolete given the rapid advancement in computer hardware and software technology, but developing computer models forces one to systematize knowledge about cognitive structure and processes in strategic decision making.

3.2 Design for the Case Study Research

As defined in Chapter 2, a mental model’s purpose is to describe, explain, or predict the form, function, or state of the represented system. If a model exists which encapsulates a subdomain of business knowledge, then descriptions, explanations, or predictions should be observed, and they should occur as one step in an overall decision
process when the corresponding subdomain is implicated in the decision. The two case studies conducted in this research are intended to establish, (1) the presence of one or more mental models related to strategic decision making, and (2) how these mental models are structured and how they contribute to strategic decision making.

A second purpose for conducting the case studies is to establish the viability of a management support system (MSS) whose architecture includes a computational model that reflects the role of a mental model in a manager’s decision making process. An MSS might include several such computational models which assist the MSS’s user to assess the status or progress of a business strategy. Not all mental models documented in the case studies may be candidates for an MSS, either because the model is not well enough understood to define a corresponding computational model, or because there is no motivation for the manager to use such an MSS. Both types of situations are considered in the case analyses.

3.2.1 Organizations Studied

3.2.1.1 Gamma Financial Services Corporation

Gamma Financial Services is an independent subsidiary of a large world-wide corporation. A majority of Gamma’s 800 employees work in jobs such as key punch operators, telephone operators, and customer service representatives. This case study was centered around the Director of Quality Assurance (henceforth referred to as the QA

---

1 The actual names of these two corporations are disguised.
Director) and his interactions with the rest of Gamma employees, especially other executives. The QA Director's efforts are mostly directed toward developing and implementing a strategy for Total Quality Management.

3.2.1.2 Alpha-Tech Corporation

Alpha-Tech is a major U.S. corporation that produces a wide range of electronic and computer products. Executives at Alpha-Tech's corporate headquarters delegate much of the strategic responsibility to business units, which in turn coordinate a related group of loosely integrated product divisions located throughout the United States. This study focuses on a well-defined strategic planning process that is promoted by corporate headquarters, but which is only adopted in "spirit" by the product division included in this case study.

3.2.2 Multiple-case, Embedded Design

The two case studies are structured as embedded designs; that is, each case uses two units of analysis at different levels (Yin, 1989). Both cases concentrate on individual managers as the primary unit of analysis. This focus seems most appropriate for satisfying a research objective that is concerned with representing mental models, which are necessarily individual phenomena.

The Gamma Financial Services case study is designed with two levels of analysis: (1) an individual manager: the Director of Quality Assurance, and (2) two strategic projects managed by the QA Director. This single manager was interviewed and
observed for a three month time period to learn about his general management style, and data was collected about each of the project’s attributes that might contribute to one or more mental models.

The Alpha-Tech Corporation study also includes two levels of analysis, but at a higher level: (1) the organizational strategic management processes, and (2) a highly integrated group of eight managers who must integrate their long-term plans for designing and introducing new products. This case is primarily used to contrast the strategic management styles in the two very different business environments from these cases and to draw conclusions related to the use of mental models in these situations.

I have partially adopted the methodological guidelines for building theories from case research as described by Eisenhardt (1989a). Several examples of her own research program for understanding strategic decision making within high-velocity firms are described in published studies (Bourgeois & Eisenhardt, 1988; Eisenhardt & Bourgeois, 1988; Eisenhardt, 1989b). Her research method includes: an initial definition of the research question and tentative a priori constructs; multiple sources of data for triangulation of evidence; and an iterative process of comparing the emerging theory with the data so as to obtain an empirically valid theory. Although Eisenhardt prescribes a multiple case study approach for verifying the theory, I combine two case studies with a computational model of the constructs.

The case studies are also used as a means to collect data for constructing a computational model. The extent to which the computational model can or cannot
accommodate these data is seen as being indicative of the computational model's value as a basis for management support systems.

3.2.3 Data Collection

Structured interviews were conducted with managers in both case studies (McCracken, 1988). A pretested question protocol, included in Appendix A, was used to guide the open-ended interviews that averaged 90 minutes each. The interviews focused on the strategic issues faced by each manager, the objectives established to resolve the issues, the methods for measuring or assessing the objectives, and the role of other persons or organizations in achieving the objectives. Relevant documents were collected whenever possible that would further explain these items.

At Alpha-Tech, two directors and two managers were interviewed at corporate headquarters, and eight managers were interviewed at one of its product divisions. The two corporate directors were selected for their unique leadership roles in orchestrating and promoting Alpha-Tech's planning processes: the Director of Corporate Planning and the Director of Corporate Quality. The two corporate managers were selected for having used Alpha-Tech's planning processes for the past two years to compile plans for their respective areas. At the product division, managers were selected to represent each of the primary functional areas involved in designing and manufacturing new computer products.

The Gamma study included several structured interviews with executives, however, the study's primary data collection vehicle was three months continuous
observation of the QA Director's projects and his day-to-day work. I sat at a desk near his office, and I attended most of his meetings in order to collect extensive data on the concepts and decision heuristics used in his strategic decision making. Whenever he made a decision, or gathered data for a future decision, I asked how or why the information would be used and who would be affected by the decision. Through this extended observation and impromptu questioning, I was able to overcome a common fault in interview data where the respondent gives socially desirable answers, either intentionally or unintentionally.

3.2.4 Data Analysis

The data analysis conducted on these case study results follows a pattern-matching logic (Yin, 1989) whereby an empirically based pattern is compared with a predicted theoretical pattern. Case study results are not subject to statistical generalization, where an inference is made about a population on the basis of empirical data collected about a sample. Case studies are not sampling units. Multiple case studies are considered like multiple experiments:

The method of generalization is 'analytic generalization' in which a previously developed theory is used as a template with which to compare the empirical results of the case study. If two or more cases are shown to support the same theory, replication may be claimed. (Yin, 1989, p.38)

The case study data are summarized in Chapter 4 by matching the evidence with the theoretical propositions that were presented in Chapter 2. Then the computational representation, described in Chapter 5, specifies how a frame-based semantic network
can be used to represent the knowledge structure in a mental model. Representational propositions are presented which summarize how the computational representation accommodates each of the theoretical propositions from the framework. The case study evidence is again matched to these representational propositions in Chapter 5, and assisted by the construction of a computational model in Chapter 6, in order to support the representation scheme's validity.

As a means of recording and summarizing the case study data, I developed a graphical representation technique to record the evolving model structure and to facilitate feedback from the participants. The graphical representation was inspired by gIBIS (Conklin & Begeman, 1988) which is an application specific hypertext system designed to facilitate the capture of early design deliberations. I am primarily interested in the typed links used in the gIBIS network. gIBIS employs three kinds of concept nodes: Issues, Positions, and Arguments, which are interconnected by nine kinds of links.

![Graphical representation for concept structure](image)

**Figure 3.2: Graphical representation for concept structure**
This idea was generalized by using a similar graphical technique to capture the concepts and relationships in the terms used by the managers interviewed. The graph is used to study patterns of concepts in a manager's mental model. A small example is shown in Figure 3.2. Three detailed examples are described with the case study results in Chapter 4. This graphical representation is used as a basis for the linguistic layer in a computational architecture presented in Chapter 5.
CHAPTER 4

STUDIES IN THE WAY OF MENTAL MODELS

Three strategic management situations, summarized from two case studies, are described as potential sources for mental models. Each situation is identified as corresponding to a reasonably well encapsulated model about a single segment of the environment, and each model is used repeatedly in similar situations. Each model helps a manager to understand the structure of a situation (i.e., the relationships among important concepts and the attributes of those concepts) and to consider alternative decisions to change the situation. Each situation is characterized by continuous information flow and by multiple, coordinated decision makers, although these case studies consider the strategic situations from individual perspectives. The first strategic management situation is taken from the Alpha-Tech study, and the remaining two situations are drawn from Gamma Financial Services.

The situations from these two case studies are analyzed for their similarities and differences with the theoretical framework for mental models. Parenthetical notes are included in these situational descriptions whenever the framework's propositions are relevant to the current discussion, e.g. (FLEXIBLE LEARNING) refers to Proposition 2.10. By this chapter's conclusion, I hope to convince the reader that these three strategic
management situations may be generalized to the theory, and thus are candidates for evaluating the computational representation presented in Chapter 5.

4.1 The Hoshin Management Process

The Hoshin Management Process at Alpha-Tech, described in the next section, was studied at two levels of the organization. Corporate-level officials were interviewed for their views on how they would like this planning process to be used, and for how they thought it was actually being used. Then functional and departmental managers from one of Alpha-Tech's product divisions were interviewed to assess how Hoshin affected their strategic planning efforts.

4.1.1 Alpha-Tech Corporate Headquarters

The Hoshin Management Process was originally adopted by Alpha-Tech's Japanese subsidiary and has been used very successfully in that location for several years. Efforts have been underway since the early 1980's to adapt Hoshin for use in the United States (Flexible Strategy). These efforts were spearheaded by Alpha-Tech's Director of Corporate Quality. I interviewed him for an overview of the Hoshin process and for his opinion on its prevalence within Alpha-Tech's business units and product divisions.

To introduce the Hoshin Management Process, a quote is included from Alpha-Tech's training material used to introduce managers to this process:
The Hoshin Management Process is a systematic process for defining key entity issues, for developing plans to ensure that these issues are adequately addressed, for reviewing progress to plans, and for making changes to the plans when required. To ensure that everyone in the entity is working toward the same end, the plans are hierarchical in nature, that is, they cascade down through the organization. ... The basic premise behind the Hoshin process is that the best way to obtain the desired results is to ensure that everyone in the entity understands the direction and is working according to plans that lead in that direction. The process includes the setting of objectives, goals, strategies, and process performance measures (PPM's), and the review of progress on the plans, all of which are part of a management Plan-Do-Check-Act (PDCA) cycle. ... The annual plan lists the objectives to be achieved as well as the goals. It also outlines the strategies that will be used to achieve the objectives. Most importantly, the plan includes PPM's for every strategy -- specific measures that will monitor, guide, or manage performance or progress on the strategies. These PPM's should essentially answer the question: How will I know whether a strategy is being successfully implemented?

The corporate culture at Alpha-Tech is such that no corporate mandates are made for requiring all business units and product divisions to use the Hoshin planning processes. The Director of Corporate Quality named one business unit where the Hoshin process was widely used, although he admitted that the Vice President of that unit had modified the process somewhat to suit his own ideas about planning (FLEXIBLE LEARNING). The Hoshin management processes is thus accepted in "spirit" rather than as policy.

One of Alpha-Tech's product divisions (not one of those noted as an active Hoshin user) was studied next in order to assess how the Hoshin process was used by functional and departmental managers.
4.1.2 Alpha-Tech Product Division

"My Hoshin plan? Hmmm, I have a notebook here somewhere... I update it once each quarter if I have time, but, quite frankly, I really never look at it. Since our general manager doesn't pay attention to it, we don't use it either." This is a typical response from one of the functional managers in this product division of Alpha-Tech. When conducting interviews below the functional management level of the division (top managers in manufacturing, R & D, marketing, administration, and quality), the lower managers were all familiar with the basic principles of the Hoshin process, but none had prepared a plan according to its specification.

Each manager that was interviewed had a mission statement and some list of annual objectives readily available, but there was no consistent structure. However, each manager did have some sort of plan and each could describe, at least in rough terms, how his or her objectives related to a two- or three-year divisional strategy. Thus, although a formal planning process is promoted within Alpha-Tech, it does not appear to have a direct influence on individual managers' mental models.

4.1.3 The Spirit of Hoshin

On paper the Hoshin process appears as a beautifully orchestrated approach for coordinating planning, albeit a bit demanding in its documentation requirements. In reality Hoshin is often rejected as too rigid and too time-consuming (HIERARCHICAL LEARNING). So, what planning procedures do managers use at Alpha-Tech? It appears that managers individuate some of the principles---basic concepts and relationships---
behind the Hoshin process. Hoshin is used in "spirit" rather than in its specific prescriptions.

Alpha-Tech has distributed extensive amounts of training material throughout the corporation that describes the Hoshin Management Process and that provides examples of completed Hoshin plans. As a consequence of this training effort, most managers in the corporation are familiar with the basic principles and terminology used in the method. So, although managers don't follow the prescription in detail, it appears that Hoshin has contributed a "planning language" for discussing the components of plans.

The elements in this planning language are organized as relationships among several basic concepts (DIFFERENTIATION):

a. understanding the Issues to be addressed;
b. defining Objectives to address the Issues;
c. specifying Goals for these Objectives;
d. defining Strategic and tactical responses for these Objectives;
e. creating specific Performance Measures for these Strategies and tactics;
f. defining roles of other Persons and Organizations.

The italicized terms are commonly used among managers, regardless of the specific form adopted for their respective plans. These common elements contribute to procedures used to assess the status or progress of goals and plans:

a. describe the status of Goals for each Objective;
b. describe the status of Performance Measures for each Strategy;
c. identify and explain which Strategies are performing poorly;
d. identify and explain coordinative opportunities among several Organizations.
e. identify and explain coordinative failures.
As outlined in the last chapter, a graphical representation was developed for charting the concepts and relationships used in a particular mental model. Based on interviews about Hoshin, and on reading the associated documentation and training material, a small representation was constructed, shown in Figure 4.1, which illustrates how the identification of issues drives the definition of objectives, goals, strategies, performance measures, and assignments to responsible individuals in the organization. 'Product' is shown detached because it may be 'affectedBy' an Issue, an Objective, or a Strategy.

Next, consider two strategic management situations drawn from Gamma Financial Services. Graphical representations will also be documented for these situations as a means to compare alternative models that might be developed in a management support system.
4.2 Total Quality Management Strategy

The focus of the embedded case study at Gamma Financial Services was to learn how the QA Director planned for the introduction of a new Total Quality Management (TQM) strategy. Gamma defines their TQM philosophy as "an integrated system of organizational management designed to achieve continuous improvement in the organization's ability to effectively and efficiently meet or exceed internal and external customer requirements." TQM is implemented with the following guiding principles:

1. Striving for excellence through continuous improvement in the products and services we deliver to both our internal and external customers.

2. Meeting or exceeding the requirements of our internal and external customers as well as the level of performance achieved by our competition.

3. Recognizing that customer/agent satisfaction is our competitive edge in the marketplace.

4. Integrating quality improvement as a normal part of our day-to-day operating philosophy that affects all departments and every employee.

5. Achieving success through the combined efforts of employees, vendors, agents and customers working as partners to improve existing products and services and in the development of new offerings.

This statement of purpose contains clear implications for the coordinated efforts among employees within the firm and for the need to understand the suppliers' and customers' roles within the business strategy. These objectives can yield an elaborate mental model representing the integrated roles of the managers, employees, and external agents (INTEGRATIVE MANDATE).
4.2.1 Malcolm Baldrige Criteria for TQM

A primary precept of quality improvement programs is that one must quantitatively measure all important aspects of the business, from manufacturing to customer service (DISCRIMINATION). The quality improvement program itself is no exception to this rule; the quality of the quality program must also be measured. The most difficult aspect of assessing service-oriented business practices is in determining how to measure them. Part of this problem has been resolved through the creation, in 1987, of the Malcolm Baldrige National Quality Award.

The Baldrige award's application criteria, summarized in Appendix B, have been widely accepted as guidelines for evaluating any organization's quality program (COMPOSITE FORMATION). These criteria are thus an important, and highly integrated, subset of the QA Director's mental model for achieving Gamma's TQM strategy. These criteria have been used to guide the design of a survey which seeks to determine the quality-related knowledge and attitudes of Gamma's employees. These criteria have been used to emphasize the need for: tightly integrating the quality strategy with the overall business strategy (STRUCTURAL); improving employee training and recognition programs; and regularly assessing customer and vendor attitudes about Gamma's quality efforts.

4.2.2 Managing a TQM Strategy

Gamma hopes to apply for the Baldrige award in 1992 or 1993. In order to complete this task, they must specifically account for all 133 'Areas to Address' documented in the application criteria. This effort might be conceived as building a
mental model which integrates the Baldrige TQM criteria with activities either underway or completed and with a functional and structural description of Gamma's organization. Minimally, a complete Baldrige application would entail completed activities addressing each 'Area to Address'.

![Diagram](image)

**Figure 4.2: Concepts and relationships in the TQM strategy**

Figure 4.2 illustrates the principle concepts and relations in the TQM strategy. The upper-left portion of the diagram captures the Baldrige criteria’s hierarchical
structure (HIERARCHICAL LEARNING). The right portion represents the organizational structure, which is integrated with a functional description of the organization. The center and lower part of the diagram represents the activities that address the Baldrige criteria and that are assigned to organizations or to their members. The 'Activity' box at the bottom of the diagram is the same concept shown in the center; it's separated due to space limitations in the 2-dimensional diagram. Activity management is a sort of resource allocation task which assigns individual people or entire organizations to strategic and tactical activities that address deficiencies in the evaluated criteria.

In the third strategic management situation, the QA Director at Gamma expands on the functional descriptions of organizations (small workgroups, departments, or entire divisions) in order to assess their productivity and to reorganize their work to focus efforts on customer satisfaction.

4.3 A Model for Organizational Productivity

The primary objective of effective productivity improvement is to reduce unnecessary and wasteful effort, not to simply speed things up. Having people do more of what they should be doing and less of what they should not be doing will yield significant rewards in both quality and productivity improvement. A mental model of an organization can assist a manager in achieving this objective. Such an organizational model is used within Gamma Financial Services as one technique for addressing its TQM
objective to improve customer satisfaction through assessing organizational structure and activities.

A mental model of an organization would need to incorporate the obvious elements, such as the organizational structure (generally a hierarchy of divisions, departments, and work groups), individuals who populate that structure, and something about the role that each person plays in producing useful output from the organizational system. The precise nature of these concepts and the relationships among them depend on the decision maker's purpose for using the mental model. In this TQM context, the model must answer questions about the costs and activities for the organization that the model represents and must assist the manager by simulating changes in the organization that attempt to improve its effectiveness at achieving customer satisfaction. The degree of improvement might be assessed by another evaluative mental model which represents the TQM strategic objectives and evaluative criteria.

4.3.1 Background

A very specific organizational modeling approach was developed by the QA Director within Gamma Financial Services, and has previously been used to analyze more than 70 other organizations over that past fifteen years (most of which were outside of Gamma). A summary of this model is captured in a quote from a Gamma brochure:

The Functional Administrative Control Technique (FACT) is a computer-assisted methodology designed specifically to address the critical issues related to indirect or overhead cost analysis and white-collar quality/productivity improvement. FACT has been designed to capture and quantify the effort and cost applied to organizational functions and
link this information with customer, management and employee attitudinal data. Factors which drive costs in organizations are identified and measured. Unnecessary costs are eliminated or reduced by examining negative cost drivers (e.g., rework, engineering changes, customer complaints, etc.). The unique combination of quantitative and qualitative information regarding organizational processes offers numerous opportunities for cost and operational improvement.

Of the benefits cited for this business modeling technique, several appear to conform especially well to the definition of mental models as describers, explainers, or predictors of real-world systems. First, "FACT is cross-departmental, permitting the examination of issues which cut across the entire organization." That is, a manager is able to examine issues vis-à-vis his or her mental model based on the FACT knowledge structure. Second, "FACT links customer, supplier, management and employee satisfaction to the work performed in the organization." So, the mental model depends on explicitly identifying relationships among its concepts. Third, "FACT uncovers why people do what they do by identifying major Cost Drivers -- the events and circumstances that cause people to do work." The organizational mental model provides explanations for indirect associations represented in the model's network of concepts and relationships. Finally, "FACT can be useful for projecting resource requirements on a functional basis." A mental model based on the FACT structure might help a manager to predict future demands on the organization.

When one considers situations where such a mental model might be useful to a manager, several alternatives are suggested: (1) developing function-based flexible budgets; (2) developing operational responses to changes in strategy; (3) assisting in the
development of tactical plans; and (4) assisting in the determination of product or service
cost.

4.3.2 Model Structure

One might ask, is the FACT model also the QA Director's mental model? The
answer is yes, and no. The FACT model symbolizes his best attempts to capture an
explicit representation of his mental model for improving an organization's productivity.
I first describe the FACT model, as a reflection of his mental model, then I consider how
his mental model goes beyond what FACT is able to capture. The next chapter describes
how I am able to capture a more natural and more complete representation of his mental
model.

The foundation of the FACT model---a response to the question: What does the
organization do?---lies in the concept of a function. A function is stated as a brief verb-
noun description of an organizational process. For example, "Resolve customer
problems" or "Coordinate corrective action." This organizational model critically
depends on a hierarchical relationship among the functions, where a response to the
question "why is a function performed?" leads from the specific to the general functions,
and a response to the query "how is a function performed?" leads downward in the
hierarchy (STRUCTURAL). See Figure 6.2 on page 166 for an example of a function
hierarchy. The level of detail expressed by these functions depends on the purpose for
which the organizational model will be used.
This FACT model has been applied to individual departments as small as five people or to entire organizations encompassing 1200 people with salary and overhead expenses exceeding $42,000,000. In small departmental models, the functions would address such processes as "Document customer problem" or "Reconcile invoices," whereas in large organizations the functions are conceptualized at a higher level: "Analyze competitors," "Establish pricing," or "Set production standards."

Depending on the manager using this model, or depending on the problem situation, some means are required to assess the contribution or the importance of a particular function to the overall organization. Three variants on organizational cost have been used to measure the level of human effort required for each function: (1) the salary cost in dollars; (2) the average hours per week or per year; and (3) the number of full time equivalent people (e.g., even if 15 people are involved in the function, their accumulated time may be equivalent to 0.75 of one full-time person). Several alternatives are again available for estimating the allocation of organizational cost to each function: (a) each employee keeps a time-log of functions performed; (b) each employee estimates where his or her time is being spent, by function; and (c) managers or supervisors estimate where time is being spent for an entire department or work group.

Recall again the primary purpose of this discussion. A manager uses some mental representation of an organization's functions, its employees, and its customers and suppliers in order to describe, explain, or predict that organization's role in strategic decisions. How will the organization respond to an anticipated environmental change (e.g., a change in product mix due to new competition)? What impact will a proposed
organization change have on the environment (e.g., devoting more resources to training new independent sales representatives)? The cost estimation procedures described above assist the model-builder in forming an accurate representation of the organization.

The construction of an organizational mental model is, at least for this Quality Assurance Director, an incremental process which always begins with the definition of a costed functional model (Differentiation). Once this portion is complete, additional information is collected and linked into the model. The principle information added at this stage represents attitudes of customers and suppliers---those who receive the outputs of the organization and those who supply its inputs. From the customers' and suppliers' perspectives: How important is each function? How reliably does the organization perform each function? Functions having high importance and low reliability should be corrected. Similar attitudes are considered for the employees who make up the organization being modeled.

This attitudinal information includes both 'hard' and 'soft' data. Hard data consists of rating functions for their importance and reliability on ten-point scales (Discrimination). Soft data includes verbal comments which provide either favorable or unfavorable opinions related to one or more functions (Inclusion). The computational representation for this mental model must accommodate both types of data.

Each function is classified as being either mission-related or non-mission-related work (Inclusion). Does performance of a function add value to the product or service? Does its performance move the organization in the direction of achieving its objectives and fulfilling its strategies? Similarly, functions may be associated with one of the four
‘Cost of Quality’ categories: internal failure, external failure, prevention, and appraisal (Crosby, 1980).

Which events or environmental situations cause costs to be incurred? Which functions are affected? These causal events are known as cost drivers (Berliner & Brimson, 1988), and they, too, are related to the organizational model by associating them with the functions and by noting whether their effect is positive or negative (e.g., does the organizational response to this cost driver add value to the enterprise or simply add cost?). This component of the organizational model is used to answer questions such as: Which cost drivers should be addressed first for maximum impact on the organization? If a cost driver can be reduced or eliminated, how will the organization be affected? How much cost will be saved?

These concepts and relationships in an organizational mental model can be partially represented in a graphical diagram, as shown in Figure 4.3. Properties of the concepts (such as a function’s cost, contribution, and reliability) are not shown here, but are described in detail along with presentation of the computational representation in the next chapter.

Cognitive complexity theory suggests that flexible complexity might be taught by encouraging decision makers to explore the components and relationships existing within a system, by permitting developmental explorations, and by de-emphasizing memorization for the system (Streufert & Swezey, 1986) (FLEXIBLE LEARNING). Memorization of a system leads to inflexible, hierarchical complexity (HIERARCHICAL LEARNING). In the case of the FACT organizational model, the process for building an
4.3.3 Decision Heuristics

The QA Director’s model is unique when compared with other common approaches to organizational cost reduction. Others methods are more unidimensional in that decisions about reorganization either (1) cut costs uniformly across all departments
in the organization, or (2) cut budget items having the highest expenses regardless of where those items impact the organizational functions.

The FACT methodology is function-centered and it explicitly associates numerous dimensions with each function in the decision model (INTEGRATIVE MANDATE). A heuristic was developed to integrate three dimensions within each function. The function’s contribution is assessed and discriminated on a 10-point scale, i.e. how much value does a particular function add to the business? A function’s reliability, as perceived by the customer, is also evaluated on a 10-point scale. The function’s cost is estimated using the methods described above. The scales for contribution and reliability are summarized in Appendix C (DISCRIMINATION).

The contribution and reliability for each function are combined by multiplying their values, yielding a new dimension called worth (LOW-LEVEL INTEGRATION; COMPOSITE FORMATION). A function’s worth is combined with its cost by using 27 decision rules to yield a recommendation for how to change that function. The recommendations range from quality-related issues ("raise reliability" or "reduce waste") to tactical and strategic issues ("expand functional area," "reduce delivery cost," or "eliminate function"). As is shown in Chapter 5, these recommendation heuristics can be embedded within a property of the function schema in the computational representation.
4.3.4 Mental Simulations

As one form of decision heuristic, the FACT model structure supports mental simulations performed within the model as an aid to decision making. A mental simulation might predict how the organization could be affected by a strategic change being considered. If a cost driver could be eliminated, or reduced by, say, 40 percent, what impact would that have on the organization? Using the relationships shown in Figure 4.3, and importance weights assigned to those relations, the following questions could be answered: Which functions are affected and by how much? Which employees or departments would receive reduced work load? How much of their time would be saved? If this change were made, and the corresponding functional costs reduced accordingly, how would this affect the functional recommendations produced by the heuristic described in the last section? Changes such as this one could be evaluated within the model before being implemented in the actual organization.

Using qualitative mental simulation in the sense described by deKleer & Brown (1983), a manager might seek explanations for ways that a particular product would be negatively perceived by retail customers. This simulation could proceed as follows: think about product X and about which organizational functions affect, or are affected by, that product; think about several common examples of large retail customers; consider negative comments that you've heard from these customers; which departments or employees were implicated in these comments?; which functions are performed by these employees, or by the employees in these departments?; note which of these functions are common with the functions affected by product X.
This example illustrates how a mental model would "run" a qualitative simulation of "walking through the organization" to generate explanations about the real-world system (organization) that it represents. Notice that the inferences generated by the model do not result from decision rules. The simulated "walk" proceeds simultaneously from two memory recollections: product X, and a general concept for retail customers. This type of search process is described as "two spreading spheres of activation" within the semantic networks discussion in Chapter 5.

The case study at Gamma Financial Services included extended observation and questioning while the QA Director constructed two complete FACT models. In Chapter 6 the data from one model is used to illustrate how such simulations may be realized in a management support system.

4.4 Summary

Both Hoshin and FACT identify management issues and both are customer oriented. However, Hoshin uses issues to drive the development of a plan (objectives, goals, etc.), while FACT uses issues as a cue for organizational change. They are complementary. Recall Hoshin's emphasis on dimensions for quality, delivery, cost, and education; compare these with FACT's functional dimensions for contribution, reliability, and cost. Both processes place strong emphasis on explicit performance measures used for assessing the status and progress of the strategic and organizational systems.
The Total Quality Management study illustrates how the mental model concept might be applied to implementing a particular strategy, rather than to managing the strategic planning process using Hoshin. The Malcolm Baldrige quality award guidelines provide evaluative criteria for assessing the quality strategy's comprehensiveness. Due to the TQM strategy's recency, its procedures were not as well defined as those in the FACT model. The concepts and relationships shown in Figure 4.2 appear to describe the QA Director's current model for the TQM strategy. Although the model is incomplete, its structure provides a base for further elaboration as the model evolves and expands in the future. As noted in several framework propositions (OPENNESS, REOPENABLE CONTEXTS, and FLEXIBLE STRATEGY), a model should always remain open to new dimensions, flexible, and sensitive to the dynamic environment.

Since the specificity of the data collected from the two FACT studies provides a better base for evaluating the computational representation, attention will be focused on the FACT organizational model in the next two chapters. Chapter 5 describes how these structural and procedural characteristics of the organizational model might be represented in a computer model. A prototype management support system that incorporates this representation is illustrated in Chapter 6 by using data from the organizational model at Gamma Financial Services.
CHAPTER 5
REPRESENTING MENTAL MODELS

A computational model encodes elements in a computer program that correspond with similar elements in a person's mental model encoded in the human mind. A computational model is defined through the use of a computational representation language, whereas a mental model is encoded via some theoretical mental representation. The common term 'knowledge representation' is avoided in this paper since it is used to refer both to mental and computational representations of knowledge.

As described at the beginning of Chapter 2, a general framework, by itself, does not constitute a theory. A predictive theory for a cognitive architecture must specify the following items:

1. The representational properties of the knowledge structures that reside in memory and their functional consequences.
2. The nature of the storage process.
3. The nature of the retrieval process.
4. The nature of method application, which breaks down to:
   a. The mechanism for identifying procedural methods.
b. The process that deposits the results of these actions in working memory.

c. The learning mechanisms by which method application affects procedural memory.

This list is adapted from Anderson's (1983) unitary theory for cognitive architecture which is based on the premise that a production system component operates on the declarative knowledge representation. In the above list, Anderson's statement has been generalized by replacing his terms 'productions' and 'production memory' with the more general references to 'methods' and 'procedural memory'. I wish to allow for a more diverse set of reasoning processes which operate on complex memory structures such as schemata or semantic networks.

The scope of this thesis cannot accommodate all of these elements of a cognitive architecture for mental models. This work focuses on the flexible integration of knowledge structures that is required for effective decision making and on the processes for storage, retrieval, and procedural activation that support this integrated knowledge base. This thesis does not address learning mechanisms and only touches on a few of the storage and retrieval processes.

The following discussion of knowledge representation always keeps an eye on its ultimate application in a management support system for strategic management tasks. In keeping this focus, it might be necessary to simplify theoretical representations in order to develop a system that is easily understood by non-technical managers, or by managers who don't have sufficient time to master a difficult, comprehensive
representation scheme. For example, the Cyc system (Lenat & Guha, 1990; Lenat, Guha, Pittman, Pratt & Shepherd, 1990; Guha & Lenat, 1990) implements an ontology for classifying objects and events that consists of 5000 categories. While this level of specificity would undoubtedly be useful in reasoning about the contents of a mental model, it would present overwhelming difficulty for a manager attempting to place a new object into this hierarchy. Indeed, the requirement for a management support system that was most often cited by the managers interviewed in the case studies was that the system should be easy to understand and its operation obvious without having to consult instructions.

Figure 4.1, Figure 4.2, and Figure 4.3 in Chapter 4 provide a relatively simple, yet apparently useful, ontology for these strategic management tasks. This chapter's objective is to present a computational representation which accommodates these simplified ontologies, which supports a set of storage and retrieval processes that can manage a reasonably large knowledge base, and which accomplishes the mental model's purposes (description, explanation, or prediction).

The first section of this chapter presents a functional architecture for mental models that incorporates the objectives noted above. Next, a set of propositions are offered for a computational representation; these propositions are matched with the theoretical propositions from Chapter 2, and are supported with evidence from the case studies. A four-level computational architecture is then presented for a computational model that incorporates the representation. The next two sections discuss the fundamental principles and inference techniques for frames and semantic networks,
respectively. These two basic representations are used to implement a computational representation for mental models. Chapter 6 describes a computational model that implements this representation. Finally, in Chapter 7, the three purposes of a mental model—description, explanation, and prediction—are reviewed, and their achievement is assessed vis-à-vis the computational representation.

5.1 A Functional Architecture for Mental Models

Anderson’s prescriptions for a cognitive architecture may be applied to the specification of a functional architecture for mental models. As described in Chapter 2, a mental model’s purpose is to describe, explain, or predict the form, function, or state of some real-world system. An agent may use one or more of these three outputs (description, explanation, or prediction) as a means to understand the decision environment and, according to Image Theory, to frame a prospective status, progress, or adoption decision. This triarchic purpose is illustrated in Figure 5.1 where the mental model, represented within the three concentric circles, interacts with the decision environment in order to serve the three types of managerial decisions suggested by Image Theory.

Mental simulation is viewed as interactions between one of these decision types and explanations or predictions generated by the mental model. Changes made in the knowledge content of the model would reflect hypothetical actions taken in the corresponding real-world system.
Chapter 2 also described the desirability of a multidimensional, flexibly integrated knowledge structure. A mental representation satisfying these desiderata is located within the inner core of the functional architecture. This knowledge is represented as a schema hierarchy for the concepts in the mental model and a propositional network relating all instances of these concept schemata. The representations for these knowledge structures are described in detail in this chapter.

A set of storage and retrieval processes must be specified which support the symbiotic relationship between the elaborated knowledge structure and the triarchic purpose of the mental model. The middle ring of the functional architecture suggests four possible processes that partially satisfy these requirements.

Figure 5.1: A Functional Architecture for Mental Models
The first process is *structure membership*, through which an instance of a concept may inherit default features from its schema and from the ancestors in the categorical hierarchy describing that schema. The members of a structure, its instances, may be enumerated, searched for particular property values, filtered to select members having a particular characteristic, etc. These schemata and their instances provide the most basic storage and retrieval processes in a mental model representation.

A second process, *direct association*, represents propositions relating two concepts. For example, the proposition “Coordinate Corrective Action performedBy John Seeley” represents an instance of the ‘Function’ schema (Coordinate Corrective Action) associated with an instance of ‘Person’ (John Seeley) via a particular relation (performedBy). Such binary associations form the elemental structure for a large propositional network that partially represents a manager’s knowledge in his or her mental model.

A third process consists of *spreading activation search*, or indirect association. This process facilitates the formation of descriptions and explanations through indirect inferences drawn from a propositional network. Inferences derived from spreading activation search might consist of previously unconsidered relationships spanning three or more concepts and associations, whereas retrieving direct associations represents recollections of previously stored propositions.

A fourth set of processes support *analogical retrieval* used for recognizing similar concepts and contexts in a knowledge base. These processes are especially crucial for framing the current decision by using dimensions from the decision environment to
retrieve similar prior decisions. The images from these previous decisions may be used as a basis for formulating the current decision; e.g., when evaluating a new goal for improving customer service response time, one should consider the success of goals and plans from earlier strategies in similar (but not necessarily identical) circumstances. This thesis underscores the importance of analogical retrieval in a mental model, but postpones a detailed specification of its operation for future research.

A functional architecture for mental models prescribes the interaction among the basic features required in a computational representation. The representation supplies a detailed specification for the knowledge structures and processes identified in the architecture. (Recall, however, that this prescription is derived from descriptive psychological theory, i.e. cognitive complexity theory and Image Theory. The model is intended to represent the actual structures and processes for individual decisions made by individual persons.)

A computational architecture for mental models, described later in this chapter, defines a four-level specification for realizing the model representation in a computer implementable form. This architecture supports efforts to implement the representation in a management support system. Prior to presenting that architecture, the computational representation is developed through both theory-driven and empirically-driven investigation.

The research method that was used to develop this thesis relied on an iterative process for balancing: (1) a theoretical framework for mental models, (2) empirical
observations from the two case studies, and (3) a computational representation. The next section describes an intersection of these three perspectives.

5.2 Representational Propositions for Mental Models

Chapter 2 presented theoretical propositions that were taken from the cognitive complexity theory and the Image Theory. A computational representation for mental models must specify a response to each of these propositions. Toward this end, a corresponding set of representational propositions are presented in this chapter. For example, propositions 2.1 and 2.2 in Chapter 2 provide motivation for propositions 5.1 and 5.2 in Chapter 5. Empirical evidence from the case studies is interspersed with the propositions. The parenthetical references included in Chapter 4 noted correspondence between the case evidence and the framework propositions. These same references are used to indicate how the representational counterparts in this chapter address those data.

Several terms used in this research require clarification. A concept may be about a tangible object such as the concept of a building or the concept of computer hardware, or about an abstract object such as a person (both mind and body, i.e. more than the tangible objects of flesh and bones). However, a concept may often be about some abstract, socially constructed idea, such as a country, an organization or a strategy. When someone describes a concept's typical properties or features, one is accessing a schema for that concept. It is assumed that every concept is described by one or more schemata. An instance of a schema is the concept of a particular object, event or
system, such as ‘Dave Carlson’ as an instance of the ‘Person’ schema or a particular strategy for increasing product quality as an instance of the ‘Strategy’ schema. The ‘Dave Carlson’ object denotes a particular abstract object in the real world.

A *property* specification in a schema modifies a concept, e.g. a person’s ‘name’ or a function’s ‘cost’. A property value may be thought of as a concept that has no other properties of its own. A *relation* in a concept’s schema is similar to a property, but it refers to some other complex concept rather than to an atomic property. For example, a relation called ‘memberOf’ may be added to the Person schema with the specification that it typically refers to an instance of the Organization schema, whereas the Person schema may include a property for ‘salary’. One could choose to define a complex schema for the ‘salary’ concept, but it is treated here as an atomic property for the sake of simplicity and computational efficiency.

A final necessary clarification is that specification of a property or relation in a schema or in its instance is distinguished from the *value* of that property or relation. A Person schema might specify that its instances typically have a ‘salary’ property, without specifying a value for that property. Similarly, a Person is typically related to an Organization, without specifying any particular organization (i.e., a value for the relation).

The representational propositions are separated into two groups which specify the structure of complex knowledge and the use of complex knowledge structures in strategy formation.
5.2.1 The Structure of Complex Knowledge

The theoretical propositions about cognitive complexity are concerned with the integrated dimensions in a manager’s knowledge structure, rather than with the specific content of those dimensions. The following representational propositions specify the nature of these dimensions as properties and relations within integrated concepts and contexts. For convenience, a figure from Chapter 2 is reproduced here, as Figure 5.2, to illustrate a process-oriented framework for describing the formation and use of computational models in strategic decision making. The propositions that follow are organized according to this process sequence.

Figure 5.2: Propositional framework: (a) Developing complex computational models; and (b) Strategy formation is dependent on the model structure.
The computational model must grow and change if it is to maintain its correspondence with a manager’s mental model. As a manager learns about a new decision-making domain, cognitive complexity theory asserts that new dimensions are typically added as non-discriminated bipolar scales; that is, an environmental stimulus is either included or excluded on the basis of that dimension. The computational representation allows inclusion-exclusion to be implemented in two ways, depending on whether the object being classified is an atomic property or a complex object having properties of its own.

Proposition 5.1 (INCLUSION): Dimensions characterized by inclusion-exclusion principles may be represented by constraining a concept property to a boolean value, by categorizing an item as an instance of a particular schema, or by establishing an unweighted relation between instances of two concepts.

As a manager learns more about dimensions that have been previously defined, values may be discriminated along a scale used to assess the dimension. For example, a manager initially learns that a Function should be assessed according to its contribution as perceived by the organization’s customers. After considering alternative ways of discriminating the contribution, he settles on six carefully distinguished degrees for assessing this dimension (these were described in Chapter 4). This dimension for contribution is represented as a property of the Function concept.

Another dimension by which a business function is assessed is the set of people in the organization who perform that function. Since a person is not understood as an atomic property of a function, this dimension is represented as a relation between two complex objects: a Function and a Person. Discriminating within this dimension is
represented by assigning a weight to this relationship. In the case of the ‘performedBy’ relation, the weight indicates the importance of this person with respect to accomplishing the function’s objectives. Conversely, the weight on the ‘performs’ relation indicates the importance of a particular function with respect to a person’s responsibilities.

**Proposition 5.2 (DISCRIMINATION):** Discrimination within a dimension is represented either as a nonboolean type constraint for a concept’s property, or as a weight assigned to a relation between two concepts.

A concept may include definitions for several dimensions (properties) that are used in its assessment. A Function may be assessed by properties for contribution, reliability, or cost. A Function may also be described by its relations with persons, organizations, products, cost drivers, etc. In a particular decision event, any one, or all, of these properties and relations may be employed. They form the typical set of dimensions for a function.

**Proposition 5.3 (DIFFERENTIATION):** Several properties and relationships may be defined for a concept, and they may be employed independently in a specific decision event, cued by the decision environment.

While differentiation is a necessary precursor to integration, excessive numbers of dimensions in a concept or excessive numbers of concepts in a context can inhibit the formation of strategic integrations by overwhelming the decision maker. The determination of when a concept or context contains excessive numbers of dimensional elements depends on the decision environment. When a crucial decision must be made quickly, or when the decision maker is overloaded with decision tasks, the threshold for context complexity is necessarily lower (Streufert & Swezey, 1986). Effective decision
Proposition 5.4 (Excessive Differentiation): A person should avoid defining an excessive number of properties and relationships when describing a concept.

The integration of dimensions may be described at two levels. Low-level integrations are performed when several diverse dimensions are combined (either weighted or unweighted) into one overall view. A simple example of this integration is illustrated in the organizational model where a function's 'worth' is determined by combining that function's reliability with its contribution. In this case, the values of the two dimensions (each value taken from the set \(\{0, 2, 4, 6, 8, 10\}\)) are simply multiplied with one another. A more complex integration is performed when 27 rules are used to integrate the function's worth with its cost in order to infer a recommendation for how that function's performance might be revised to improve the organization's productivity. This integration is still considered as low-level since it uses properties that are restricted to one concept.

Proposition 5.5 (Low-Level Integration): A conceptual integration is defined by the set of properties associated with one concept. A property's value may be derived from weighted or unweighted combinations of other properties within this concept.

A second level of integration, high-level integration, is performed when the properties of several different concepts are integrated through relationships defined among two or more concepts. I refer to these sorts of integrations as contexts. This thesis only suggests a few of the processes for forming contexts, vis. the decision
framing process in Image Theory. Indeed, "the context in which decisions occur gives them meaning" (Mitchell & Beach, 1990, p. 10). Thus, a context might consist of the persons, strategies, and business functions associated with the resolution of a particular customer's complaint. The properties of these related concepts are employed when forming a decision in this context.

**Proposition 5.6 (HIGH-LEVEL INTEGRATION):** A contextual integration is defined by a propositional network formed by relations among two or more concepts. A relation from one concept to another may be weighted for the importance and/or the accessibility of the associated concept.

The notions of propositions and propositional networks are discussed in detail later in this chapter. Briefly, a proposition is expressed as a binary relationship associating a subject with an object. A weight may be attached to this association which specifies the importance and/or the accessibility of the object. These weights are described later in this chapter under the topic of inference through association.

**Proposition 5.7 (STRUCTURAL):** The properties of concepts, and the relationships among concepts, can be either fixed or flexible in their abilities to support alternative model configurations.

If a concept definition is seen as necessary and sufficient, then the model will be used in an inflexible manner. Multiple, differentiated dimensions (properties) may be used in hierarchical integration, but they are used in an identical configuration for each decision related to that concept. Flexible integration, on the other hand, views the concept description as a typical set of dimensions that may be used either partially or wholly. Further, atypical properties may be added to concept instances for unusual situations.
Proposition 5.8 (Hierarchical Learning): A model based on a learned algorithmic computation that uses a fixed set of propositions from one or more concepts will lead to hierarchical integration.

Successful decision making in a complex, dynamic environment depends on the decision-maker's ability to flexibly integrate the most relevant concepts and properties for each decision event. Fixed problem formulations, such as those found in most mathematical models, are only successful over the long-run in steady-state environments (Streufert & Swezey, 1986, p. 75).

Proposition 5.9 (Steady-State): Fixed concept descriptions (i.e., rigidly integrated propositions and relations) will provide a successful representation only in steady-state environments.

Flexible complexity in decision making is facilitated by exploring the properties and relationships in a model. It's unlikely that all concepts, properties, and relations should be employed in every decision event, unless the model is very small or underdeveloped. Thus, the user needs some facility for trying out alternative configurations as he or she seeks to conduct mental simulations of decision scenarios. This thesis focuses on the knowledge structure and on procedures for manipulating it as a resource for mental simulation. It is not a goal of this research to examine the control procedures used in the simulation exercises themselves.

As the mental simulation proceeds, the knowledge structure should support these pro forma scenarios without committing the values to its permanent storage. It may, however, be desirable to store several of these scenarios as decision alternatives for future reconsideration. This possibility will be discussed in a later proposition.
Proposition 5.10a (FLEXIBLE LEARNING): *The model should support exploratory configurations without committing changes to the permanent structure.*

A corollary requirement to the exploratory nature of mental models is the need to de-emphasize memorization of the model structure. If the model must be memorized before it can be understood and used in decision making, then the model will lead to hierarchical integration rather than to flexible integration.

Proposition 5.10b (FLEXIBLE LEARNING): *Memorization of the model should be de-emphasized by making the conceptual structure evident to the user.*

The ability to make a model’s conceptual structure evident to the user is at least partially determined by the user interface in a management support system, although the representation must support the model features described above. Chapter 6 presents several such aspects of the user interface that are implemented in the current prototype system.

Making a computational model’s conceptual structure evident to the user is also partially determined by the degree to which the computational model reflects the structure and semantics of the user’s own mental model. If the computational and mental models do not correspond with one another, then the user is required to learn another mental model that reflects the computational model, and map this secondary mental model onto his primary model. A central goal of this research is to allow the user to construct a computational model that adequately reflects his primary mental model, both in its structure and in its manipulatory procedures.
Flexibly integrated contexts form the basic representational structure for the decision framing processes described by Image Theory. Image Theory contains no proposals for the cognitive processes used to recall the frames of similar prior decisions, but emphasizes the importance of matching either identical or similar strategies. The primary goal of this research is to specify a knowledge structure capable of supporting these recollections; future research will investigate the processes for analogical retrieval by which these prior contexts are recalled (Thagard, Holyoak, Nelson & Gochfeld, 1990; Vosniadou & Ortony, 1989). One can appeal to the cognitive complexity theory for a conceptual description of the internal structure of these decision frames. The propositions contained in this chapter present a representation for the complex knowledge structures suggested by complexity theory.

In the model for improving organizational productivity, the decision maker may entertain several hypothetical scenarios for reorganizations: eliminating a cost driver, laying off the members of a department, canceling a product, etc. Each of these scenarios can be represented as a context which includes the affected concepts and propositions from the knowledge base, and which holds property values that are limited to the scope of that context.

**Proposition 5.11 (Reopenable Contexts):** Decision framing is supported by using environmental cues to recall and to re-evaluate similar contextual integrations that represent prior decisions or alternative scenarios.
A computational representation that supports a cognitively complex manager should be open to additional dimensions and its knowledge structure should permit the reorganization necessary to accommodate these new dimensions.

**Proposition 5.12a (OPENNESS):** The knowledge structure supports additions or reorganizations through adding typical properties and relations to an existing schema, or through adding atypical properties and relations to instances of that schema.

Proposition 5.12a (OPENNESS): The knowledge structure supports additions or reorganizations through adding typical properties and relations to an existing schema, or through adding atypical properties and relations to instances of that schema.

A Person schema includes typical properties for name, salary, etc., and includes the typical relations that a person is a member of one or more departments in the organization and that a person performs a set of functions. Other schemata are similarly defined for Organization and for Function. However, this model’s user may have atypical information about a specific person, John Seeley. It is known that John previously was a member of two other departments in the organization and is likely to have an atypical amount of knowledge about those departments. Thus, a special relation should be established between John Seeley and these other departments, but this relation is not added to the Person schema.

Reorganization of a manager’s knowledge structure is not limited to adding properties and relations to existing concepts. As a manager develops a mental model for improving an organization’s productivity, he or she discovers that the concept of a ‘Function’ is generalizable to many situations and, thus, begins to elaborate the fledgling ‘Function’ schema.

**Proposition 5.12b (OPENNESS):** The knowledge structure supports additions or reorganizations through adding new concepts by: (1) abstracting typical properties of a new schema from several instances; (2) splitting an existing concept into two new concepts and, possibly, creating
a superordinate concept which describes their commonality; (3) collapsing two concepts into one; and (4) promoting an atomic property from an existing schema to a new complex concept, and defining a relation between the original schema and the new concept's schema.

The manager may initially work with a set of specific descriptions, such as "Resolve agent problems and issues," "Develop solutions," and "Coordinate corrective action" before he realizes that these descriptions can be subsumed under a general concept for Functions. A Function schema is created with properties and relations that are typical among the initial instances. Similarly, existing schemata may be split or collapsed as a manager learns about the differentiation of these new concepts.

What may originally be understood as an atomic property may later be elaborated into a complex concept having properties of its own. For example, the original function-like descriptions may initially be understood as simply a list of activities performed by an organization, and may be stored as values of a property attributed to the Organization concept. No other properties are attributed to these activities. However, as the manager learns that functions can be independently assessed with respect to their contribution, reliability, and cost, the Function property is promoted to an abstract concept with a set of properties. Typical relations are established that associate Functions with Organizations and with Persons. As the manager learns more about the typical traits of these Functions, additional properties and relations may be added to the concept, according to Proposition 5.12a.
5.2.2 Complexity and Strategy Formation

In earlier incarnations of the organizational model, functions were only evaluated on two properties: cost, and the number of persons who performed that function (called 'fragmentation'). Later, functions were assessed by considering their 'cost of quality' category. Most recently, the properties for reliability and contribution were added. According to Proposition 5.5, these forms of within-concept structures are referred to as conceptual integrations and are classified as low-level integration.

High-level integrations, on the other hand, consist of contexts that refer to multiple conceptual integrations. The complexity of the function concept was expanded to include relationships to: comments made by suppliers, customers, and employees; cost drivers; and affected products. Similarly, a higher-order decision heuristic which considers a function's worth, together with the strategic priority of the products affected by that function, would entail a high-level integration.

The cognitive complexity of a manager's decision-making style varies with the uni- or multi-dimensionality, the differentiation, the discrimination, and the integration exercised in the knowledge structure. The degree of complexity actually employed in a particular decision event depends on the environmental demands for time and information volume. A computational representation that accurately reflects these stylistic differences should allow for degrees of complexity in its structure.

**Proposition 5.13 (COGNITIVE STYLE):** For simplistic styles, a concept is evaluated on only one property. For structurally differentiated decision styles, several properties are considered from one or more concepts, but only one property is used to form a strategy.
For flexibly integrated styles, several decision contexts are formed, where each context includes a weighted combination of properties from several concepts. One of these contexts is selected as the current strategy, and the other contexts are retained as contingencies.

A strategy, according to Streufert & Swezey (1986), must integrate multiple dimensions in its formation. An additional requirement is added here that a strategy must integrate the properties of multiple concepts; thus, strategy formation is necessarily a case of high-level integration. This is particularly evident in the prescriptions of Total Quality Management where customer satisfaction should be considered in all business strategies.

**Proposition 5.14 (INTEGRATIVE MANDATE):** *Strategy development must involve the creation of decision contexts which integrate properties from several different concepts in order to adequately reflect the decision environment.*

Because a complex decision environment requires a manager to be cognizant of many dimensions while forming or evaluating a strategy, several low-level integrations are often used to reduce the cognitive overload while the higher-level strategy is considered. For example, a decision heuristic is used in the organizational model whereby a recommendation is generated through consideration of a function's cost, contribution, and reliability. This low-level integration---formed within one concept---is then used as the basis for further high-level decisions which consider strategic priorities, organizational restructuring, etc.

**Proposition 5.15a (COMPOSITE FORMATION):** *Composites may be formed by integrating several properties into higher-level properties as an aid in reducing cognitive overload.*
High-level strategic decisions require the consideration and reconsideration of current and potential events, conditions, and outcomes. A mental simulation process may be used to form and evaluate successive hypothetical or factual scenarios. These scenarios are created as integrated contexts, which are then evaluated by procedural heuristics that are embedded within the properties of the concepts defined for the decision-maker's model. The mechanisms for embedding these procedures are discussed under the topic of frames and inference later in this chapter.

**Proposition 5.15b (Composite Formation):** Mental simulation is used to evaluate potential system states by constructing integrated contexts for both factual and hypothetical scenarios.

A strategy must be continually reevaluated in light of the dynamic decision environment. Image Theory classifies these decisions into status and progress evaluations. Image Theory's compatibility test is applied in the decision, and a change is recommended only when the context associated with the current strategy is incompatible with environmental variables. The theoretical propositions presented in this section suggest that concepts are defined, and integrated into contexts, to facilitate this reevaluation.

**Proposition 5.16 (Flexible Strategy):** Strategy development and implementation must be sensitive to continuous information flow in the decision environment. A manager should periodically reevaluate the compatibility of active contexts (through status and progress decisions) with respect to environmental cues.

Each manager's mental model includes semantics for relating names of categories and names of relationships to corresponding objects or associations in the real-world. See, for example, Figure 4.1, Figure 4.2, and Figure 4.3 in Chapter 4. The
computational representation for mental models should retain, whenever possible, the semantics used by each individual, while providing a procedure for mapping these semantics onto a common structure and inference process. A generalized representation scheme is described next which attempts to achieve this goal.

5.3 A Computational Architecture

An architecture for computational models (as opposed to a theoretical cognitive architecture for mental models) is defined which specifies the components of the representation scheme and their realization in a computational model. A four-layer architecture is presented in Table 5-i. This architecture provides a means for implementing the knowledge structures in a computational model that ultimately support the storage, retrieval, reasoning, and learning processes.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linguistic Layer</td>
<td>- Natural language semantics</td>
</tr>
<tr>
<td>Epistemological Layer</td>
<td>- Defines structure of concepts</td>
</tr>
<tr>
<td>Object Management Layer</td>
<td>- Properties and relations</td>
</tr>
<tr>
<td>Implementation Layer</td>
<td>- Frames, slots, facets, and demons</td>
</tr>
<tr>
<td></td>
<td>- Smalltalk/V286 data structures</td>
</tr>
</tbody>
</table>

Table 5-i: Computational Architecture for Mental Models
5.3.1 Linguistic Layer

The top layer, the Linguistic Layer, corresponds most closely to the language that agents use to describe the conceptual components of their mental models, the properties of those concepts, and the relations or associations among concepts. The linguistic layer is specified using commonsense terminology and natural language semantics; that is, the names or descriptions of individual concepts, properties, and relations refer to the same real-world objects as typically understood by that agent. For example, Figure 4.3 illustrates these semantics for object categories and relations in an organizational model. The ability to generalize this linguistic level representation among multiple agents depends on the degree of shared knowledge among these agents with respect to these semantics.

Several processes can be described for operating on semantic network structures that represent this linguistic level. In particular, this work uses a spreading activation search that is especially useful for providing commonsense explanations about inferences resulting from indirect associations among individual concepts at this level of the architecture (Quillian, 1967, 1968; Collins & Loftus, 1975). A general specification for the internal structure of individual concept nodes is, however, provided by the next level of the computational architecture.

5.3.2 Epistemological Layer

The second layer of the architecture, the Epistemological Layer, directly addresses the structure of the concepts employed in a agent's mental model. (A similar
epistemological layer was described by Brachman (1985) and Lenat & Guha (1990).) Brachman defines Concepts as, "formal objects used to represent objects, attributes, and relationships of the domain being modeled. A Concept is thought of as representing an intensional object, and no Concepts are used to represent directly extensional (world) objects" (p. 207). Generic Concepts represent classes of individuals by describing typical members of the class and Individual Concepts are represented by relationships with more general Concepts. Brachman notes that objects in the world have complex relational structure, and thus, they cannot be usefully taken as atomic entities or mere lists of properties. A Concept must therefore account for this internal structure as well as for the object as a holistic entity.

This computational architecture specifies the epistemological layer as a general structure for schemata that describe the *typical* properties and relations of a concept. The epistemological layer is limited to specifying the internal structure of schemata; no particular relations or properties are defined. The structure presented here is somewhat simpler than that proposed by Brachman. My intent, as stated earlier, is to provide a realistic architecture for management support systems using current technology.

A schema, in this specification, consists of two primary substructures: properties and relations. A property is an atomic element that modifies a concept but has no properties of its own, whereas a relation refers to some other complex concept within the model. The epistemological layer provides general facilities for creating properties and relations in a schema.
A property may specify its characteristics through one or more facets. The particular facets supported by this layer are limited to those specified in the object management layer, but, at a minimum, the ability to create facets for type restrictions, default values, and activated procedures should be provided. A 'type' facet limits the property's value to a specified data type, e.g. a string, an integer, or a floating point number. A 'default' facet supplies a value for the property if no explicit value is available. Various activated procedures, or demons, may be provided, such as 'ifNeeded', 'ifAdded', 'ifAccessed', or 'ifRemoved' (Fikes & Kehler, 1985; Thagard, 1984).

<table>
<thead>
<tr>
<th>Function</th>
<th>properties:</th>
<th>Agent</th>
<th>properties:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>cost</td>
<td></td>
<td>name</td>
</tr>
<tr>
<td></td>
<td>fee</td>
<td></td>
<td>address</td>
</tr>
<tr>
<td></td>
<td>contribution</td>
<td></td>
<td>phoneNumber</td>
</tr>
<tr>
<td></td>
<td>reliability</td>
<td></td>
<td>relations:</td>
</tr>
<tr>
<td></td>
<td>definition</td>
<td></td>
<td>performs Function</td>
</tr>
<tr>
<td></td>
<td>recommendation</td>
<td></td>
<td>hasComment Comment</td>
</tr>
<tr>
<td>relations:</td>
<td>performedby Agent</td>
<td></td>
<td>hasInterest Issue</td>
</tr>
<tr>
<td></td>
<td>howPerformed Function</td>
<td></td>
<td>is_a Social Category</td>
</tr>
<tr>
<td></td>
<td>whyPerformed Function</td>
<td></td>
<td>responsibleFor Strategy</td>
</tr>
<tr>
<td></td>
<td>causedBy Cost Driver</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>affectsProduct Product</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>hasExpense Expense</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>hasComment Comment</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Person</th>
<th>properties:</th>
<th>relations:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>salary</td>
<td>aKindOf Agent</td>
</tr>
<tr>
<td></td>
<td>hours/week</td>
<td>memberOf Organization</td>
</tr>
</tbody>
</table>

|                     | manages Organization |

Table 5-ii: Concept schemata for Organizational mental model
In the context of a model for organizational productivity, concept schemata are defined for Function, Agent, and Person, among others. Table 5-ii illustrates partially specified schemata for these concepts; default values may be included for some, or all, of the properties and relations. Note that Function includes a relation with Agent, and Agent includes a reciprocal relation with Function. The relations defined in these schemata form the basis for a propositional network at a model's linguistic level. The propositional network is constructed from instances of these concept schemata.

5.3.3 Object Management Layer

The object management layer supports the schema structure definitions in the epistemological layer, i.e. identifying the role of an object's attribute as either a property or a relation, supporting the creation of facets, and managing the execution of procedures embedded within the facets. These objects are also used to implement the instances of a schema, e.g. an individual person denoted by the name 'Dave Carlson'. The computational model uses a frame representation language for implementing this layer.

The object management layer is more closely aligned with implementing the computational model than with specifying the representation scheme. Since object-oriented design is often compared with frame-based representation as complex data/knowledge structures, the phrase 'object management' is used in the layer's name as a general reference to either object-oriented design or to frames.

In object-oriented design, objects are defined by a class which corresponds roughly with a schema. Classes, like schemata, may be organized in a hierarchy where
variables and methods are inherited from more general classes. However, at least in current implementations of object-oriented programming languages, classes are more akin to necessary and sufficient definitions than to descriptions of typical features. Instances of classes have no facility for specifying atypical variables or methods. My choice of frame-based representation for implementing the structure of concepts in mental models is described in detail later in this chapter.

5.3.4 Implementation Layer

The implementation layer determines the physical level data structures used to implement the high-level structures in the object management layer. The management support system developed as part of this research, called SPRINT, implements a computational model using a frame-based semantic network. A hypertext interface is implemented for manipulating the model (Halasz, 1988). The system uses the Dictionary collection class of Smalltalk/V286 to implement classes for frame, slot, and semantic network.

5.4 Schemata, Frames, and Inference

Schema theory describes a general psychological theory for complex knowledge structures and for how these structures influence memory and reasoning processes (Rumelhart, 1975; Rumelhart, 1980). Frame theory was developed under the aegis of artificial intelligence (Minsky, 1975) as similar to, but more specific than, the early
schema theory (Bartlett, 1932). Minsky is credited with the active revival of schema theory among cognitive psychologists (Brewer & Nakamura, 1985). Frame theory is used in this thesis to specify a computational representation for mental models and to implement a computational model.

This section presents an informal, intuitive description of frames. A representational formalism is presented later in this chapter using a set-theoretic notation.

5.4.1 An Overview of Frame Representation

Many discussions of frame representation use the term ‘frame’ in an ambiguous manner, referring both to the general definition of a concept’s typical features and to particular instances of that concept. The term ‘schema’ is used here for referring to these general concept frames which are defined using a frame representation language. A particular instance of a schema (e.g., ‘Dave Carlson’ as an instance of the Person schema) is also defined using a frame representation language, and the instance frame inherits typical properties and relations from the general schema frame. Thus, when the term ‘schema’ is used while describing the computational representation, it refers to a schema frame described in a frame representation language.

A frame, for either a schema or an instance, defines properties and relations as slots. A slot, in turn, defines a series of facets which specify that slot’s characteristics and its role in the overall frame. See Table 5-iii. It is the detailed specification provided by these facets that principally distinguishes frame theory from schema theory. Schema
theory, by itself, is not sufficiently detailed to guide the implementation of a computational model. Frame theory has been a major research topic—both basic and applied—in the artificial intelligence field since Minsky's introduction in 1975.

A frame for the 'Function002' schema in an organizational model might be partially specified as shown in Table 5-iv. (The symbol 'Function002' is a conventional notation for indicating a particular instance of the 'Function' frame.) The first two slots, for 'properties' and for 'relations', specify the role of other slots within the frame; that is, determining those slots which represent typical properties of the concept, versus those which represent typical relations between this frame and other frames. These first two slots each contain only one facet which contains the value of the slot. The 'name' property includes another facet that restricts the slot's value to the data type 'String'. The 'cost' property includes a facet called 'ifNeeded' that computes the slot's value when it is queried, but not known. If the 'cost' slot also contained a 'value' facet, then the
frame: Function002
  slot: properties
    facet: value (name, cost, contribution, reliability, ...)
  slot: relations
    facet: value (is_a, howPerformed, whyPerformed, performedBy, causedBy)
  slot: name
    facet: type (String)
    facet: value ('Coordinate Corrective Action')
  slot: cost
    facet: type (Float)
    facet: ifNeeded (computeFunctionCost)
  slot: contribution
    facet: type (Integer)
    facet: value (8)
  slot: is_a
    facet: value (Function frame)
  slot: howPerformed
    facet: type (Function frame)
    facet: value (Function007, Function012, Function076)

Table 5-iv: A partial Function frame

ifNeeded procedure would not be activated. Other similar types of facets for activated procedures (generally referred to as "demons" in the frame literature) would include 'ifAdded' and 'ifRemoved'.

Note that the ifNeeded facet creates behavior similar to a backward-chaining inference process: if a value is not available, a rule (ifNeeded procedure) determines how to infer or compute the missing value. If the ifNeeded demon attempts to access a value in another frame which is undefined (in this example, the procedure may request the unknown value of the cost slot in another subordinate Function frame related via the 'howPerformed' relation), its ifNeeded demon will also fire. The ifAdded facet creates
behavior which is analogous to a forward-chaining inference process: whenever a value is set, a rule (ifAdded procedure) determines which other assertions (frames/values) should also be tested or modified. Like the ifNeeded demons, the ifAdded demons may cause a chain-reaction through the network of associated frames.

5.4.2 Frames and Flexibility

The importance of flexibility in an agent’s knowledge structure and reasoning processes was introduced through cognitive complexity theory, and has been emphasized throughout this thesis. An agent needs flexibility in her knowledge structure so that she might accommodate changes in the concept hierarchy as she learns about a new domain or as she refines her existing knowledge base. Flexible structures also permit an agent to define integrated contexts that are responsive to decision framing processes. Frames provides several features that make them especially well-suited to this flexibility requirement.

When considering a particular person, John Seeley, it may be important for the decision maker to know that Seeley had previously been employed by the U.S. government IRS department. This information is useful because Seeley is a member of an organization that is preparing plans for a large sales contract with the IRS. However, in general, the decision maker does not care about the previous employers of the organization’s members. The typical properties of a person would include name, address, phone number, etc. A concept schema in a model should support both typical
and necessary properties and relations, while allowing atypical properties and relations in the description of individual instances.

At the object management or implementation layers of the computational architecture, flexibility must be allowed for modifying existing schemata. For example, as a person learns more about managing his organization’s productivity, he might decide that a causality relationship between a ‘Function’ and a new concept called ‘Cost Driver’ was both important and typical of all functions. The computational architecture must support this representational modification of a mental model.

A new typical relation might be added to the Function schema, without changing existing instances of individual functions. However, all instances will immediately inherit the presence of the new relation along with any default values and activated procedures associated with the new slot. In most object-oriented programming languages, modifying the class definition requires one to destroy all existing instances in order to add a new variable. Relational databases require one to unload and reload the database when adding or removing attributes. Most frame representation languages allow slots to be added without affecting the instances of that frame. An instance of a schema frame is just another frame that is related via a particular slot, e.g., ‘John Seeley’ ISA ‘Person’. Thus, each frame can be manipulated quite independently from the other.

In contrast, a relational database system lacks the flexibility required for unstructured, dynamic tasks such as strategic management. A database schema defines the necessary and sufficient attributes for each entity—primary keys are necessary, while the entire set of attributes defined for a database system are implicitly sufficient. This
latter sufficiency condition restricts the flexibility of database systems in tasks where there is limited a priori knowledge of the problem domain (necessary or sufficient conditions are not known), or where the individual instances of the schema (tuples in the case of database systems) have unique or atypical properties.

5.5 Semantic Networks and Inference

The use of semantic networks as a knowledge representation formalism has a long, albeit somewhat ill-defined, history in AI literature. Quillian (1967, 1968) is generally credited with originating the idea of "semantic networks." He intended to develop a theory of the structure of human long-term memory and to create a computer model which utilized the theory to perform complex, memory-dependent tasks. The tasks which Quillian addressed were primarily concerned with storing the meanings of commonplace words.

Although Quillian's intent was to model the semantics of English words using network structures, similar representations were soon being used to model many nonsemantic subjects, e.g. physical object structures. Virtually every network-like formalism that has appeared since Quillian's work was published has been labeled by someone as a "semantic net" (Brachman, 1985). Brachman describes a generalized network-like formalism which he calls a "Structured Inheritance Network," or SI-Net. A SI-Net has a fixed set of node and link types which provide a basis for a fixed and well-defined interpreter and an explicit semantic theory for the formalism.
For this thesis' purposes, several features of semantic network representations are described which are useful for describing the mental models of managers. Brachman's recommendations are intentionally simplified in order to develop a computational architecture that is immediately applicable to implementing management support systems. I do, however, follow the spirit of his prescriptions for a network formalism with an "epistemologically explicit" representation.

Basically, a semantic network consists of nodes, interconnected by various kinds of associative links; thus, it is similar to an ordinary dictionary. Links from one concept node (word) point to other concepts (words) which collectively form a definition of the original concept. This characterization, by itself, is a bit oversimplified in that it ignores both the different roles played by nodes and the meaningful semantics contributed by particular links. The following discussion of an integrated representation for structured knowledge addresses these deficiencies.

Quillian intended his semantic memory model to serve as a general inferential representation for knowledge. An intersection search is conducted as a spreading activation, breadth-first search of the nodes surrounding two concepts. The search fans out by following links from the original two concept nodes, until a point of intersection is found between these "spheres of activation." The resulting path would indicate a potential relationship between these two concepts. Thus, implicit relationships may be inferred from the explicitly defined network.
Prior to delving into the detail of an algorithm for spreading activation search, frame and semantic net representations are merged into a comprehensive representation for structured knowledge.

5.6 Structured Knowledge

The nodes in the networks implemented in this research project represent both intensional concepts (schemata) and denotations of extensional objects (instances of schemata). Relationships among schemata, including hierarchical inheritance, describe the intensional knowledge structure of the domain being modeled; relationships between a schema and an instance describe that instance’s role within the integrated knowledge structure; and relationships between two instances describe associations among particular real-world objects. Figure 5.3 illustrates this distinction for several schemata (shown as ovals), instances (rectangles), and properties (circles).

The internal structure of each node is represented using the facilities of the epistemological layer in the computational architecture. At the object management layer, each node is implemented as a frame, where a frame’s slots point to other frames, thus forming a semantic network. The frames also support Brachman’s (1985) “conceptual coat rack” whereby metaconceptual procedures can be hung onto the concept definition as demon procedures, within either a schema or an instance.

In Figure 5.3, note the proposition that “Coordinate Corrective Action” is performedBy “John Seeley.” Each of the three components in this proposition—subject,
relation, and object—is an instance of a schema: Function, performedBy (Binary Relation), and Person, respectively. A hierarchy of schemata is also shown where Agent subsumes both Person and Organization. The instance-schema or schema-schema inheritance relations are shown as heavy, single-ended arrows pointing toward the parent schema.

Each schema in Figure 5.3 defines several typical properties that are inherited by its subschemas or its instances. In addition, the Function schema defines a typical relation, called 'performedby', that has Function and Agent as its subject and object, respectively. The figure shows a performedBy schema (oval) that is a subschema of Binary Relation, and a performedBy instance (rectangle). The 'performedBy' instance
specifies that the importance weight has the value 0.15, while the accessibility weight inherits the default value of 1.0 from the Binary Relation schema. These weights are used in directing the spreading activation search (Collins & Loftus, 1975).

An instance of a schema may inherit default values of properties, or the instance may override those defaults with instance-specific values. For example, ‘John Seeley’ overrides the default value of ‘hours/week’ (40) with the value 25. Other properties, e.g. Agent name, have no default values.

The ‘recommendation’ property in the Function schema illustrates how a domain-specific expert heuristic may be embedded within the model definition. This property includes an ifNeeded facet that implements a 27-rule decision procedure. Thus, whenever the recommendation is requested for a Function instance, the ifNeeded procedure is inherited from the Function schema, and the recommendation’s value is derived by using property values specified for contribution, reliability, and cost.

The instance frame ‘Coordinate Corrective Action’ adds an additional property, ‘secondary recommendation’, that is not typically defined for each Function. If at a future date this model’s user discovers that most Function instances include this same additional property, then its definition could be added to the Function schema, perhaps with a default value or with an activated procedure used to derive its value when required. This promotion of an atypical property in an instance to a typical property in the schema illustrates one way in which learning processes might be realized in the knowledge structure (cf. Propositions 5.12a and 5.12b).
Another learning process in a frame-structured knowledge base would occur either when a schema is split into two subschemas or when two previously unrelated schemas are partially consolidated by creating a new parent schema that subsumes both of the original schemas. The second of these two alternatives was used to create the Agent-Person-Organization hierarchy. That is, the model's user realized that a concept called 'Agent' could be understood to provide a more abstract description of both persons and organizations.

It is important to note at this time that the computational representation makes no attempt to automate these learning processes. However, the knowledge structure should accommodate the changes that correspond to these examples of learning. These sorts of changes are best accommodated when the computational architecture, especially at the object management and implementation layers, provides flexible manipulation of the knowledge structure.

5.6.1 Propositional Networks

The simplest proposition in this model is stated as: Subject relation Object. Such binary propositions reflect assertions made about instances of two concepts. For example, using the organizational model semantics from Figure 4.3 on page 92, several assertions can be made: Jim Danials performs Develop Solutions, or Agent Services managedBy Pam Goodman. Other propositions may be asserted that ultimately link these first two propositions: Jim Danials memberOf Retail & Financial Institutions, and Retail
Financial Institutions hasParentOrganization Agent Services. Thus, it may be inferred that Jim Danials ultimately works for Pam Goodman.

A propositional network, then, is formed by the interlaced set of propositions having common subjects and objects. A very small example of such a network is shown in Figure 5.4. Although the proposition that “Jim Danials performs Develop Solutions” may be made as an independent statement about the decision environment, prior assertions stored in the model enable inferences to be made about Jim Danials’ role in the organization.

Since it is known that, for example, ‘Agent Services’ is an Organization and ‘Develop Solutions’ is a Function, one immediately recalls additional information about each of the subjects and objects in this network. The schematic knowledge structure contributes conceptual information to the instantiated content in the propositional network.

Although some authors do not distinguish intensional knowledge in schemata from instances of those schemata in a semantic network representation (instances are simply “terminal nodes” of a network), it is often conceptually beneficial to distinguish structure from content in this representation for mental models. One can appeal to cognitive complexity theory for a similar distinction. The structure of a manager’s mental model can be discussed without mentioning content, and one can make assertions about “Developing Solutions” and “Agent Services” without specifically mentioning their concept schemata. Representations are developed in this thesis for mental model structure, and that structure is exemplified through content from the case studies.
Figure 5.4: Propositional Network

5.7 A Representational Formalism

A formal definition of the computational representation can be described in a set-theoretic notation\(^2\). A standardized notation is used where an uppercase italic letter denotes a set and a lowercase italic letter denotes an element of a set. The expression \(\bigcup_{d} D\) refers to the generalized union of \(D\), i.e. the set that results from forming the union of all the sets in \(D\).

This formalism, used to describe the structured knowledge in a computational model, consists of a directed graph described by:

\(^2\) Suggestions for this formalism were drawn from Shastri (1988, 1989), Pollock (1990), and Abiteboul & Hull (1987).
\[ \Theta = \langle \Phi, \Psi, \Gamma \rangle \]

where \( \Phi \) represents the sets of structural elements in the model, \( \Psi \) represents sets that define the ranges of the functions in \( \Gamma \), and \( \Gamma \) is the set of functions defined on \( \Phi \) and \( \Psi \). The sets of structural elements in \( \Phi \) are further defined as the n-tuple of

\[ \Phi = \langle C, R, P, F \rangle \]

where \( C \) is the set of concepts, \( R \) is the set of relations, \( P \) is the set of properties, and \( F \) is the set of facets.

The function set, \( \Gamma \), is defined as:

\[ \Gamma = \langle \lambda, \delta, \tau \rangle \]

where \( \lambda \) encompasses a family of structural functions that are defined on the sets in \( \Phi \), \( \tau \) is the schematic function, and \( \delta \) encompasses a family of value assignment functions defined on \( \Phi \) and \( \Psi \).

The range of each function \( \delta_i \), denoted as \( R(\delta_i) \), is taken from one or more of the sets in \( \Psi \):

\[ \Psi = \langle C, D, \Lambda \rangle \]
where $C$ is the same concept set referred to in $\Phi$, $D$ is the set of domains defined on atomic object types (such as, reals $\mathbb{R}$, integers $\mathbb{I}$, booleans, characters, strings, graphical bitmaps, etc.), and $\Lambda$ is the set of procedural functions. The procedural functions represent the activated procedures, or demons, attached to a frame's slots, e.g. ifNeeded, ifAdded, ifRemoved, etc.

5.7.1 Structural Functions

The following structural functions are defined:

1. $\lambda_R : C \rightarrow P(R)$, i.e. $\lambda_R$ is a mapping from $C$ to the power set of $R$

2. $\lambda_p : C \rightarrow P(P)$

3. $\lambda_f : (C \times P) \rightarrow P(F)$

$(\forall c)(c \in C)$, $\lambda_R(c)$ yields the subset of $R$ that consists of relations that are applicable to $c$. In a mental model of an organization based on the FACT method, $\lambda_R$(Person) would yield \{is_a, memberOf, manages\}, and $\lambda_R$(Agent) equals \{is_a, hasKinds, performs, hasComment\}. $\lambda_R$ describes the knowledge structure of $\Theta$ by capturing the relations that one concept has with other concepts.

$(\forall c)(c \in C)$, $\lambda_p(c)$ yields the subset of $P$ that consists of properties that are applicable to $c$. For example, $\lambda_p$(Function) might equal \{definition, cost, contribution, reliability, recommendation\}. Properties are distinguished from relations in that
properties have a multitude of facets describing their role in defining the concept, whereas relations have only concepts as their values.

\((\forall (c, p))(c \in C \& p \in P), \lambda_F(c, p)\) yields the subset of \(F\) that consists of facets that are applicable to \((c, p)\). A property's facets, for a particular function, might be exemplified by \(\lambda_F(\text{Person, avgHoursPerWeek}) = \{\text{type, value, default, ifAdded}\}\).

### 5.7.2 Schematic Function

A crucial aspect of schema/frame theory is a frame's ability to inherit relations and properties from its ancestors. A particular relation, typically called 'IS-A', provides an efficient "hard-wired" inheritance path. Earlier in this chapter, the distinction between schemas and instances was emphasized, although the same inheritance relation is used for instance-schema or schema-schema inheritance. These schematic inheritance structures in \(\Theta\) are defined by:

\begin{align*}
(1) \quad \tau : C & \rightarrow \{ \text{True, if } c \in C \text{ is a schema} \\
& \text{False, if } c \in C \text{ is an instance} \}
\end{align*}

\(2) \quad R_n, \text{ the inheritance relation, is a partial ordering, where } R_i \in R.\)

\((\forall c)(c \in C), \tau(c)\) returns a set containing 'True' if the concept \(c\) is a schema, otherwise it returns 'False'. A generalized notation is used here for the inheritance relation, \(R_n\), rather than simply referring to IS_A. As will be seen in Chapter 6, a management
support system implemented with this representation may choose a more intuitive name for \( R_i \), such as ‘filedIn’ which corresponds to the filebox metaphor described there.

### 5.7.3 Value Assignment Functions

Finally, the assignment functions determine the values of each concept’s relations and facets. A relation’s value is a set of special _relational concepts_ that act as intermediaries between two or more other concepts. Relational concepts may be defined for binary relations, ternary relations, and so forth. A facet’s value is a set taking its elements from one of the sets: concepts, domains, elements of a domain, or procedural functions.

First, values are assigned to each concept’s relations:

\[
\delta_R: (C \times R) \rightarrow P(C_R).
\]

Values are then assigned to the facets of each concept’s properties:

\[
\delta_p: (C \times P \times F) \rightarrow D \cup P(d) \cup P(\Lambda), \text{ where } d \in D.
\]

A particular property’s facet may map to a single domain, a subset of one domain, or a subset of the procedural functions. Relational concepts, as a subset of all concepts, also may have other concepts assigned to their properties’ facets:

\[
\delta_{PR}: (C_R \times P \times F) \rightarrow P(C),
\]

where:

(a) \( C_R \subset C, C_N \subset C, \text{ and } C_N \cap C_R = \emptyset \); 

(b) \( \forall c \in C_R \Rightarrow \lambda_R(c) = \{R_p, R_p^{-1}\} \).
i.e. for relational concepts, the only valid relations are the inheritance relation and its converse; and

(c) $\Lambda = \{f | (\exists A)(\exists B)(A,B \in (C \cup (\cup_G D)) \& f(A) = B)\}$,

where $\cup_G D$ is the generalized union of $D$, i.e. the set that results from forming the union of all the sets in $D$.

Alternatively, these value assignment functions may be expressed as:

1. $\delta_R(c,r) \in C_R$, where $c \in C$ and $r \in \lambda_R(c)$;

2. $\delta_p(c,p,f) \in D$, or $\delta_p(c,p,f) \in d$, or $\delta_p(c,p,f) \in \Lambda$,

   where $c \in C$, $p \in \lambda_p(c)$, $f \in \lambda_f(c,p)$, and $d \in D$; and

3. $\delta_{PR}(c,p,f) \in C$, where $c \in C_R$, $p \in \lambda_p(c)$ and $f \in \lambda_f(c,p)$.

Examples of these value assignment functions would include: $\delta_R(\text{Function073, performedBy}) = \{\text{BinaryRelation017, BinaryRelation042}\}$, i.e. two relational concepts each link this function instance with one other concept; $\delta_p(\text{Function073, name, value}) = \{"\text{Coordinate Corrective Action}\}$, i.e. the function name contains a string object; or $\delta_p(\text{Function, recommendation, ifNeeded})$ would yield a subset of $\Lambda$ with procedures for inferring a function's recommendation. Note that these procedural functions are in a

---

3 A notational convention common in frame-based representation literature is followed here where, for example, 'Function073' refers to some particular instance of the Function schema.
property of the Function schema, so they will be inherited by all Function instances. A more complete example is presented at the end of this section.

5.7.4 Procedural Control

As described in detail by Thagard (1984) and Shastri (1988, 1989), frame representation is not simply a notational variant of first-order predicate logic. This is principally because frame representation includes procedural control mechanisms that are not a part of FOPL. The "hard-wired" nature of inheritance was already described by including the relation $R_I$ which is a partial ordering defined on the concepts.

Another aspect of this procedural control is the built-in default value mechanism which returns the contents of a property's value facet first, if available, else returns the default facet's contents, else returns the result of evaluating the ifNeeded facet's procedure. If no value is returned by this series of events, then the same process is invoked again in the frame's ancestral schemata, accessed via $R_I$. This facet ordering is described generally by a relation:

$\eta$ is a simple ordering defined on the set $F$

For this example, $\eta = \{(\text{value}, \text{default}) , (\text{default}, \text{ifNeeded}) \}$. Other relations could be defined for $\eta$.

5.7.5 An Example of the Representation Language
For an extended example, consider a very small subset of the organizational model. This example focuses on descriptions for relational concepts. As footnoted above, terms such as 'performs011' refer to some particular instance of the 'performs' schema.

\[ C_R = \{ \text{BinaryRelation}, \text{is}_a, \text{hasKinds}, \text{performs}, \text{performedBy}, \text{performs011} \} \]

\[ C = \{ \text{Person}, \text{Function}, \text{Person046}, \text{Function017} \} \cup C_R \]

\[ R = \{ \text{is}_a, \text{hasKinds}, \text{performs}, \text{performedBy} \} \]

\[ P = \{ \text{name}, \text{address}, \text{cost}, \text{subject}, \text{object}, \text{converse}, \text{importance}, \text{accessibility} \} \]

\[ F = \{ \text{value}, \text{type}, \text{default}, \text{ifNeeded}, \text{ifAdded}, \text{ifRemoved} \} \]

The 'BinaryRelation' concept, a schema, is described as:

\[ \tau (\text{BinaryRelation}) = \{ \text{True} \} \]

\[ \lambda_R (\text{BinaryRelation}) = \{ \text{is}_a, \text{hasKinds} \}, \text{where } R_i \text{ is 'is}_a \text{ and } R_i^{-1} \text{ is 'hasKinds'} \]

\[ \lambda_P (\text{BinaryRelation}) = \{ \text{subject}, \text{object}, \text{converse}, \text{importance}, \text{accessibility} \} \]

\[ \delta_R (\text{BinaryRelation, is}_a) = \{ \text{Relation} \} \]

\[ \delta_R (\text{BinaryRelation, hasKinds}) = \{ \text{performs} \} \]

\[ \delta_P (\text{BinaryRelation, importance, type}) = \{ x | x \in \mathbb{R} \land 0 \leq x \leq 1 \} \]

\[ \delta_P (\text{BinaryRelation, importance, default}) = \{ 1.0 \} \]

\[ \delta_P (\text{BinaryRelation, accessibility, type}) = \{ x | x \in \mathbb{R} \land x \geq 1 \} \]

The 'performs' concept, a subschema of BinaryRelation, is described as:
\[ \tau(\text{performs}) = \{\text{True}\} \]

\[ \lambda_R(\text{performs}) = \{\text{is}_\text{a}, \text{hasKinds}\} \]

\[ \lambda_P(\text{performs}) = \{\text{converse}\} \]

\[ \delta_R(\text{performs}, \text{is}_\text{a}) = \{\text{BinaryRelation}\} \]

\[ \delta_R(\text{performs}, \text{hasKinds}) = \{\text{performs011}\} \]

\[ \delta_{PR}(\text{performs}, \text{converse}, \text{value}) = \{\text{performedBy}\} \]

\[ \delta_{PR}(\text{performs}, \text{subject}, \text{type}) = \{\text{Person}\} \]

\[ \delta_{PR}(\text{performs}, \text{object}, \text{type}) = \{\text{Function}\} \]

The 'performs011' concept, an instance of the 'performs' schema, is described as:

\[ \tau(\text{performs011}) = \{\text{False}\} \]

\[ \lambda_R(\text{performs011}) = \{\text{is}_\text{a}\} \]

\[ \lambda_P(\text{performs011}) = \{\text{importance, subject, object}\} \]

\[ \delta_R(\text{performs011}, \text{is}_\text{a}) = \{\text{performs}\} \]

\[ \delta_P(\text{performs011}, \text{importance}, \text{value}) = \{0.8\} \]

\[ \delta_{PR}(\text{performs011}, \text{subject}, \text{value}) = \{\text{Person046}\} \]

\[ \delta_{PR}(\text{performs011}, \text{object}, \text{value}) = \{\text{Function017}\} \]

Similar descriptions could be written for the concepts: Person, Function, Person046, and Function017.
Note in the above descriptions that the 'performs' concept inherits five typical properties, without values, from the 'BinaryRelation' concept. The 'performs' concept also inherits 'type' facets which constrain the value facets of 'importance' and 'accessibility', and it inherits a default facet for the 'importance' property. Similarly, 'performs011', an instance of 'performs', inherits a value for its 'converse' property from the 'performs' concept.

5.8 Inference Through Association

Spreading activation search, briefly introduced as an inference mechanism within semantic network structures, provides a means for inferring implicit knowledge from indirect associations in propositional networks. A manager might wish to determine which manager(s) are best able to address the Cost Driver "Shipment errors" shown in Figure 5.4. That is, one wishes to infer an indirect relationship connecting an instance of Cost Driver with an instance of Person, such that the Person instance is also known to have a 'manages' relation with some organization.

This indirect relationship might take the form of a path through the network consisting of alternating instances of non-relational concepts and relational concepts:

\[ \text{path} = (c_0, cr_0, c_1, cr_1, \ldots, cr_{n-1}, c_n) \]

where

\[ c_i \in C_N; \quad cr_i \in C_R; \]
\[ cr_i \in \{ x \mid (\exists r)[r \in \lambda_R(c_i) \& x \in \delta_R(c_i, r)] \}; \text{ and} \]
\[ c_j \in \delta_{PR}(cr_{j-1}, \text{object, value}), \text{ for } j \geq 1. \]

For example, (Shipment Errors, causes, Resolve Agent Issues, howPerformed, Develop Solutions, performedBy, Jim Danials, memberOf, Retail & Financial Institutions, hasParentOrganization, Agent Services, managedBy, Pam Goodman, manages, Agent Services).

Spreading activation proceeds as a breadth-first search from the initial node by traversing all of its relations to their respective nodes. Each of these "activated" nodes similarly spreads activation to each of its relations and connected nodes. By initiating a single search, a large number of paths are created that span the network. Obviously, in a fully-connected network the spreading activation would proceed until every node in the network was activated. An algorithm was implemented in this research which controls the search process through the use of four parameters. These parameters are summarized in Table 5-iv, Table 5-v. Note that indefinite looping is eliminated by preventing nodes from being traversed more than one time. Once a path has passed through a node, no other path can include that node.

### 5.8.1 Goal Condition

The goal condition determines when a search has reached a successful conclusion. For example, when a node has been reached that is an instance of Person and that has a 'manages' relation to some other node. Three possible goal conditions are:
Four parameters:
  - Goal condition
  - Constraint condition
  - Decay rate
  - Threshold value

Search terminates when:
  - Goal is true
  - Constraint is true
  - Activation energy drops below threshold value
  - Node has already been traversed

Table 5-v: Algorithm for Spreading Activation

1. \( c_g = c_n \), i.e. the last concept node in the search path equals the goal concept;
2. \( r_g \in \lambda_R(c_n) \), i.e. the last concept node has a relation \( r_g \); and
3. \( x_g \in \delta_P(c_n, p_i, \text{value}) \), i.e. the last concept node has a value in its property \( p_i \) that equals the goal value, \( x_g \).

5.8.2 Constraint Condition

The constraint condition specifies when the search should logically be terminated as unsuccessful. A condition may be encountered where continuation of the search is unlikely to produce useful conclusions. These conditions may be determined by analyzing the domain being modeled and enumerating the paths (relations only) that, according to human expertise, are most likely to produce meaningful inferences (Cohen & Kjeldsen, 1987; Kjeldsen & Cohen, 1987). A constraint could then be specified to eliminate paths that are not among this endorsed set. The constraint condition can also support generalized heuristics that reduce the search output to paths that are most likely
to be useful to the user of a management support system. Three examples of general heuristics are described here that might usefully constrain many different models.

First, because both schemas and instances are nodes in the semantic network representation, the spreading activation search can traverse relations to schemata as well as to instance nodes. It is generally not useful to continue searching beyond a schema, since all of its instances would be activated in the next stage of the search. Thus, a constraint condition could be specified to terminate the search when:

(1) $\tau(c_n) = \{\text{True}\}$, i.e. the last concept is a schema.

Another useful search constraint is that a search should be terminated when a path contains both a relation and its converse. For example, from Figure 5.4, the path

\[
\langle \text{Develop Solutions, performedBy, Jim Danials, performs, Coordinate Corrective Action, performedBy, George Burton, performs, \ldots}\rangle
\]

is a natural outcome from an unconstrained search. This type of search path can easily be eliminated by terminating the search when:

(2) $\forall cr [cr \in \text{path} \& cr^{-1} \in \text{path}]$

A third example of a constraint condition would be to terminate the search whenever the last concept has a "high fan-out," i.e. the concept has a large number of relations leading to other concepts. The magnitude of this constraint may need to be experimentally determined for each model, or for each family of models having similar semantics. Cohen & Kjeldsen (1987) show that, in their network, paths ending with nodes having branching factors between 8 and 10 maximize the recall of useful concepts.
while keeping the nonuseful paths relatively low. By including nodes having a branching factor greater than 10, the number of nonuseful paths increases at a much higher rate than the additional useful paths. Thus, a heuristic might be written to prune paths that encounter nodes having more than 10 branches. This condition might obtain for general concepts or for a concept that participates in many propositions. This constraint could be used in lieu of constraint (1) in order to include schemata having few instances, while excluding those schemata that have a large number of instances. The condition can be expressed as:

\[(3) \{x \mid (\forall r)[r \in \lambda_R(c_0) \& x \in \delta_R(c_0, r)]\} \text{cardinality} > m\]

5.8.3 Decay Rate and Threshold Value

Cohen & Kjeldsen (1987) constrain spreading activation by enumerating path endorsements and by terminating at nodes with high fan-out. They also terminate any path that exceeds four relations in length. While such an absolute criterion may work in their application domain, it seems desirable to employ a more general process that is also responsive to relations having varying strengths or weights.

The decay rate and threshold value provide a general mechanism for terminating searches that have exhausted their activation energy. Each search begins with an initial activation energy of \(A_0\) units assigned to concept \(c_0\). For each relation traversed, the remaining energy is reduced by a fraction specified by the decay rate, \(\alpha\). Each path of
the search is terminated when its remaining activation energy drops below the threshold value, \( T \). That is,

\[ A_i = (1-\alpha)A_{i-1}, \quad \text{stop when } A_i < T, \]

where \( A_i \) is the activation energy of \( C_j \) in the search path and \( A_{i-1} \) is the energy of the previous node, \( C_{i-1} \).

However, this initial formulation does not account for variable weights assigned to a relation that associates two or more concepts. Collins & Loftus (1975) suggest that these relational weights may have two purposes: importance (or criteriality) and accessibility. The importance of a link is a number indicating how essential that link is to the meaning of the concept. For example, the importance of a particular person for successfully performing a function, or the importance of a function for accomplishing a strategy. Links may also have different accessibility (i.e., strength or travel time). The accessibility of a relational link "depends on how often a person thinks about or uses a property or concept. Whether criteriality [importance] and accessibility are treated as the same or different is a complex issue, but network models allow them to be treated either way" (p. 409).

In the formulation presented in this thesis, importance and accessibility are properties of relational concepts. Their values are incorporated into the activation energy update as follows:

\[ A_i = (1-\alpha)\text{Imp}_{i-1}\text{Acc}_{i-1}A_{i-1}, \quad \text{stop when } A_i < T, \]

where:
(a) \( \text{Imp}_{i-1} = \delta_p(cr_{i-1}, \text{importance, value}), \) i.e. the importance property of the relational concept connecting \( c_{i-1} \) with \( c_i \);

(b) \( \text{Acc}_{i-1} = \delta_p(cr_{i-1}, \text{accessibility, value}); \)

(c) \( 0 \leq \text{Imp} \leq 1; \) and

(d) \( \text{Acc} \geq 1. \)

If both weights are assigned default values of 1, then their values have no effect on the spreading activation process. When the importance of a relation is less than 1, the activation energy will decrease more quickly and the search will terminate sooner. If a relation’s accessibility is greater than 1, then the activation of node \( c_i \) is boosted in opposition to the constant decay rate and to the node’s fractional importance. If a relation’s importance were 0.2 and its accessibility equal to 5, then the two weights would cancel one another’s effect. Thus, frequent access of a low-importance relation will, at least temporarily, lengthen paths via a relation that would otherwise be shorter than average.

The accessibility weight was not tested when constructing computational models for this thesis, but it is provided, as suggested by Collins & Loftus (1975), as a learning mechanism where networks could be responsive to repeated access to a particular relation. No method is suggested for changing a relation’s accessibility value, either for increasing its value based on frequent access or for decreasing its value over time to simulate forgetfulness. Investigation of this learning potential is postponed to future research.
The decay rate and threshold value mechanism described here can easily provide the same termination behavior as Cohen & Kjeldsen by assigning values to these two parameters and by ignoring the link weights. For example, if $A_0=1$, $\alpha=0.25$ and $T=0.1$, then the search will terminate after traversing 9 links. These values were selected to provide a path length sufficient to span the models tested in this work.

The next chapter includes two examples of the spreading activation search as implemented in a prototype management support system. Exploration of alternative values for the search parameters and their impact on inference in a computational model for strategic management is postponed until a much larger model has been constructed. However, the objectives of this thesis have been satisfied by specifying and implementing procedures for manipulating mental models that are based on the theory of cognitive science and by demonstrating how these procedures are able to accommodate models based on case study evidence.

5.9 Summary

Presentation of this representation began with a functional architecture for mental models. A mental model's triarchic purpose---producing descriptions, explanations, and predictions about a corresponding real-world system---is supported by mental representations for a concept schema hierarchy and a propositional network, and by a repertoire of storage and retrieval processes that have a symbiotic relationship with the
knowledge structure. This functional architecture provides the necessary support for framing a manager's status, progress, and adoption decisions.

A complementary computational architecture guides development of a computational model by specifying four layers in a system's design. The linguistic layer reflects the concepts and relations in a manager's language used to describe the domain being modeled. An epistemological layer is concerned with the internal structure of individual concepts. The object management layer supports definition of attributes and activated procedures within a concept, and is implemented as frames, slots, and facets in my prototype system. The lowest level, the implementation layer, specifies physical level data structures---Smalltalk/V286 objects in the SPRINT management support system.

By developing a set of representational propositions, it was shown that this representation can be matched with, and generalized to the earlier theoretical framework. Case study data was used to illustrate this correspondence.

The next chapter demonstrates how this representation can be implemented in a management support system and how the system design can make the representation's flexibility available to the user. Two models are constructed in this system using data from the case studies.
A practical computer system must be designed which embodies the computational representation and which allows a person to build a computational model of the essential elements in his or her mental model for a strategic management task. A prototype system, called SPRINT (Strategic Plan and Resource INTegration), was implemented that uses a frame-based representation as an integrative structure for: (1) a semantic network, including spreading activation search and inference; (2) a production rule inference mechanism based on ‘ifAdded’ and ‘ifNeeded’ demons embedded in each frame’s slots; and (3) a hypermedia interface for creating and accessing the model. The semantic network supplies the fundamental node/link representation of hypermedia, with the addition of semantic information carried by the link names.

The knowledge structure was described in terms of frames and semantic networks in the last chapter. This chapter expands on the role that a hypermedia interface plays in the SPRINT system and describes several key features in the system design. Data from two strategic situations described in Chapter 4 are used to illustrate SPRINT’s knowledge structure and its hypermedia interface for creating and manipulating a model’s contents.
6.1 Hypermedia Interface for an MSS

As a brief introduction to the purpose and use of hypermedia, Halasz (1988) summarizes:

Hypermedia is a style of building systems for information *representation* and *management* around a network of multi-media nodes connected together by typed links. Such systems have recently become quite popular due to their potential for aiding in the organization and manipulation of irregularly structured information in applications ranging from legal research to software engineering.

Among hypermedia systems, the NoteCards system, developed by a research team at Xerox PARC (Halasz, 1988; Halasz, Moran & Trigg, 1987), was most influential in the SPRINT system design.

The NoteCards system is designed around two primitive constructs, *notecards* and *links*. A notecard is an electronic form of a 3x5 paper notecard which is generalized to contain any digital media: text, bitmap image, etc. Links are used to interconnect individual notecards into a network of related concepts. A link is a typed, directional connection between a source and a destination card, where the type is a user-chosen label signifying the meaning of the relationship. *Fileboxes* are specialized notecards that can be used to organize or categorize large collections of notecards. A filebox may contain any type of notecard, including other fileboxes. Fileboxes were designed to help users manage large networks of interlinked notecards by building a hierarchical categorization of notecard clusters.
Halasz (1988) reviews seven issues to be considered in designing the next generation of hypermedia systems. One of these issues is particularly relevant to incorporating intelligence into a mental model representation. Unlike expert systems, hypermedia systems do not include inference engines that actively derive new information and enter it into the network. In cases where computational ability is added to hypermedia, there is a clear separation between the network and the computational engine. Cards and links may be added to the network, or the network may be searched for information, but the procedural knowledge is not integrated with the passive network structure. Halasz summarizes, "A merging of concepts from frame-based systems into the design of hypermedia systems would be a sensible way to approach the integration of hypermedia with rule-based, truth-maintenance, and other computational engines."

This approach is taken in an integrated representation for SPRINT.

Although hypermedia systems expand the user's accessibility to a wide range of information which may have very little semantic structure, an already severe problem of selecting the most appropriate information from available knowledge or databases may become worse. However, if the conceptual structure of the hypermedia system reflects the user's personal mental model, then selection and navigation become more intuitive for the user. Links from the node currently being viewed may serve as memory prompts for related concepts; that is, the explicit model represented in the system reflects the user's implicit mental model.
This is not a dissertation on human-computer interaction; however, a few of the key features in SPRINT's design are described to illustrate my future research in this area.

6.1.1 A Schematic Memory Metaphor

The SPRINT system generalizes the NoteCards filebox/notecard structure into a more general metaphor that corresponds with the theoretical schema-instance structure in human memory. A SPRINT filebox is used to create a schema definition, as shown by the ovals in Figure 5.3. A filecard corresponds with an instance of a schema, e.g. a rectangle in Figure 5.3. Metaphorically, the system user can store a filebox in another filebox as a way of representing a schema hierarchy, or a filecard is placed into a filebox as a representation for a schema's instance. Thus, whenever a person using SPRINT creates a new filecard (instance) and files it in a filebox (schema) that filecard automatically inherits the characteristics defined for that filebox. By creating a filecard for 'Dave Carlson' and storing it in the 'Person' filebox, the 'Dave Carlson' node "knows" its typical relations and properties without having them explicitly specified.

6.1.2 Graphical Display and Filtering

A graphical display provides SPRINT's users with a flexible mechanism for viewing the knowledge base structure. A user can select any node within the semantic network, choose a relation which connects that node to one or more other nodes, and view a graph of the hierarchical tree created by all nodes related to the originally selected
node via the chosen relation. A tree greater than one level deep is created whenever the secondary nodes have relations by the same name. This is especially true of recursive relations, such as a Function's howPerformed or whyPerformed relationships shown in Figure 4.3. Numerous examples of the graphical display are shown later in this chapter; a graphical tree of organizational functions is shown in Figure 6.2. Scan those figures briefly at this point to facilitate the upcoming discussion.

The SPRINT system's users are aided in controlling and scanning the graphical display by employing three display filters, singly or in concert. These display filters are created and stored as filecards in the 'Filters' filebox, which is accessed like any other filecard in the current model.

Name filters. As each node in the network is displayed, its title, or a summary description, must be extracted from the knowledge base. By default, each node includes a string value in a slot named 'title', or an 'ifNeeded' facet in the 'title' slot. Also by default, a filebox (i.e., a schema) is shown surrounded by square brackets (e.g., [Function]). However, the user can modify this default behavior by creating a name filter.

If a name filter is defined and activated, the output from this filter's procedure is used in the graphical display rather than using output from the default title procedure. For example, a name filter was activated in Figure 6.5 to cause the link weight to be appended to each title. That is, each node displays the importance weight for the link
connecting its parent node to itself. This name filter is currently written such that if the link weight is equal to 1 (the most common default value), then no weight is displayed.

**Color filters.** It's often helpful to provide additional information about the nodes in the graphical display by modifying their color, depending on one or more of the node's attributes (relations or properties). Unfortunately, color does not show very well in these black & white figures.

In Figure 6.10, a color filter was activated to display any filecards filed in the [Areas to Address] filebox as blue if they do not have links to any cards in the [Activity] filebox, otherwise display them as black. This is an extremely useful filter, allowing the user to quickly scan the display of Baldrige areas to determine which of those areas are not yet addressed by some activity. When all 'Areas to Address' are black, they are all covered by one or more activities.

Similarly, a color filter was activated in Figure 6.4 to display managers as blue and non-managers as black. Although it's not obvious in the figure, Judy Danforth and Juan Perez are blue, indicating that those 'Person' filecards have a 'manages' relation to an 'Organization' filecard.

Several color filters can be simultaneously activated. If two filters attempt to modify the same filecard, then the last filter will override the earlier filter. An analogous conflict resolution mechanism is used for conflicting name filters.
Selection filters. As each level of the graphical tree is generated, all children of a node are selected by extracting the values in the relation being displayed. In Figure 6.5, all nodes in the ‘performs’ relation are displayed as children for the Person at this tree’s root. Similarly, all nodes in the ‘howPerformed’ relation are displayed, recursively, for each Function in Figure 6.2.

It is often useful to restrict this display to a subset of these child nodes. Rather than showing all functions for the person in Figure 6.5, a selection filter may be activated to display only those functions having a relation weight greater than 0.1. This filter might be especially useful in large networks.

6.1.3 Spreading Activation Search Filters

SPRINT provides a means for the system user to create a set of predefined filters that specify the four parameters required by the spreading activation search algorithm described in Section 5.8. Like the graphical display filters, the search filters are created and stored in a filebox within the current model.

The next two sections provide a very brief description for how SPRINT might be used to create computational models representing two of the strategic situations reviewed in Chapter 4. Both examples should be viewed from the QA Director’s perspective at Gamma Financial Services as representations of his mental models and should be considered as potential management support systems to augment his strategic decision making.
6.2 An MSS for Organizational Modeling

When creating a computational model which attempts to implement a representation of a manager's mental model, I begin with the linguistic level. That is, what are the major concepts and relationships, described in the manager's own terms? Such a description is shown in Figure 4.3 for the QA Director's organizational model.

An initial model is established in SPRINT by creating a set of fileboxes that correspond to the concepts. Figure 6.1 shows a hierarchy of fileboxes in SPRINT's graphical browser. Menus are provided for specifying the typical properties of filecards in each filebox and the typical relations between filecards in different fileboxes.

Filecards may then be added to each filebox, and the filecards linked to one another to reflect associations in the domain being modeled. A group of Function filecards are shown in Figure 6.2, with the 'howPerformed' relation creating a hierarchy in this dimension. Figure 6.3 shows a more detailed subset of these functions in the graphical browser, plus a new window which allows the user to examine a particular filecard's contents. In this case, general textual notes are shown which define one function's scope.

If the manager wishes to view another perspective on one concept (e.g. ask the question, "Who is the function 'Coordinate Corrective Action' performed by?"), then the mouse is clicked on that filecard and the 'performedBy' relation is selected on the resulting menu. Figure 6.4 shows the display of this new perspective on the organizational model. This screen displays Judy Danforth in the color blue, indicating
that she is a manager. To find out which functions Danforth performs, select this filecard and choose the ‘performs’ relation. Figure 6.5 shows this new point of view.

Finally, some questions asked of the model must be explained via inferred relations. Since persons are only directly associated with the most detailed functions in the multi-level function hierarchy, the user cannot directly display which persons are related to a high-level organizational function. Spreading activation search may be used for this purpose. First, a general predefined search filter is selected that terminates (has a goal parameter) when a filecard is encountered that is filed in the ‘Person’ filebox. The function ‘Determine Corrective Action’ is selected with the mouse, and a search is initiated from a menu. The search results are presented in a new window, as shown in Figure 6.6.

A person, Juan Perez, is selected from the list of persons found, which causes an explanation for this indirect relationship to be displayed in the window’s lower pane. The explanation, a path through the network that connects the original function with this person, is based on the model semantics from Figure 4.3. The model semantics were implemented in the SPRINT system by initially establishing the fileboxes and relationships. The ‘Activation energy’ shown with the explanation indicates how far the search propagated in the network; i.e. how much energy remains from the initial 1.0 units of energy. A similar explanation is available for each person shown in the upper-right pane.

This brief example illustrates how the computational representation might be implemented in a management support system and describes several useful features for
creating and manipulating a model. For contrast, the next section provides a similar example from the TQM strategy.

Figure 6.1: Fileboxes in an organizational model
Support Agent Base

Resolve Agent Problems/Issues
  - Receive Communication
  - Determine Corrective Action
  - Communicate Solutions
  - Handle Inquiries
  - Resolve BFP Issues
  - Follow-up Problem/Issue

Change Agent Status
  - Document Status Change
  - Change Remittance Status
  - Change Outlet Number
  - Change Agent Name
  - Follow-up Status Change

Train Agents
  - Receive Training Request
  - Execute Training
  - Document Training (Status)

Support Hybrid Products
  - Process Special Requests
  - Monitor Performance
  - Process Paid-Side
  - Advise Daily Funding
  - Process Blocks

Monitor Agent Base
  - Call Agents
  - Document/Analyze Agent Responses
  - Administer Operations - General

Figure 6.2: Function hierarchy
Determine Corrective Action

- Gather Information
- Write Information Request
- Obtain Verbal Information
- Develop Solutions
- Document Solutions
- Coordinate Corrective Action

Figure 6.3: Notes on a function filecard
Figure 6.4: Who performs this function?
Figure 6.5: What other functions does she perform?
**Figure 6.6: Who performs this function?**

- **Function**: Determine Corrective Action
  - How Performed: [Function] Gather Information
  - How Performed: [Function] Retrieve/Pull Documents
  - Performed By: [Person] Perez, Juan

<table>
<thead>
<tr>
<th>Function</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determine Corrective Action</td>
<td>Dennis, Linda</td>
</tr>
<tr>
<td></td>
<td>Commons, Mary</td>
</tr>
<tr>
<td></td>
<td>Chen, Kenneth</td>
</tr>
<tr>
<td></td>
<td>Maxwell, Karen</td>
</tr>
<tr>
<td></td>
<td>Higgins, Diane</td>
</tr>
<tr>
<td></td>
<td>Perez, Juan</td>
</tr>
<tr>
<td></td>
<td>Dennis, Linda</td>
</tr>
<tr>
<td></td>
<td>Commons, Mary</td>
</tr>
<tr>
<td></td>
<td>Chen, Kenneth</td>
</tr>
</tbody>
</table>
6.3 An MSS for TQM Strategy

A manager held responsible for implementing a Total Quality Management strategy might benefit from a management support system that assists him or her in tracking this multi-year effort. As in a system representing the organizational model, a computational model representing the TQM strategy begins with a semantic description of the domain, as shown in Figure 4.2. The concepts from this diagram are implemented as SPRINT fileboxes; see Figure 6.7 for an illustration of the filebox hierarchy. Recall that these fileboxes correspond with schemata in a computational representation and that each filebox includes definitions for its typical relations and properties.

SPRINT's graphical browser is used to examine the filecards stored in the Malcolm Baldrige 'Category' filebox in Figure 6.8. From this view of the model, the third category, 'Plan Quality Strategy', can be selected in order to view its 'Items' and 'Areas to Address'. This view, shown in Figure 6.9, aids the manager in understanding what is required of him in order to achieve a world-class TQM program within his organization.

Each Area to Address must be addressed by one or more quality-related activities. SPRINT's color filters provide an excellent method to quickly review the status of all areas. Figure 6.10 displays areas in blue if they do not have activities related to them; i.e. they have not yet been addressed. (Once again, these black & white figures do not illustrate the color. About half of the Areas to Address in the figure are shown in blue.)
Many indirect associations cannot be viewed in the graphical display, but can be inferred through spreading activation search. For example, if a manager wanted to know, for a particular Item, which persons in the organization were somehow involved in addressing that Item, he could not obtain this information through filecards directly associated with that Item. No Person filecard is directly linked to an Item filecard. Using the same generic search filter employed in the organizational model, a search may be initiated from any filecard with a goal to find Person filecards. After initiating a search from the Item 'Define Planning Process', a window like that shown in Figure 6.11 is displayed with the search results.

The search window displays a list of all persons found that satisfy the search parameters (goal, constraints, decay rate, and threshold). By selecting a person in the upper-right pane, the lower pane displays an explanation for the Item--Person association, using the model semantics implemented via the filebox structure.

The above TQM model description is a very brief introduction to a few of the many uses for a management support system in this domain. These examples show how the computational model might represent a few of the key concepts, relations, and properties in the QA Director's mental model for implementing this strategy. Other aspects of the model were implemented in this prototype system and many others could be implemented in the future. Chapter 8 describes my plans for future research on management support systems for Total Quality Management strategy.
Figure 6.7: Fileboxes in the TQM model
Create Quality Leadership
Collect and Analyze Information
Plan Quality Strategy
Utilize Human Resources
Assure Quality of Products & Services
Document Quality Results
Assure Customer Satisfaction

Figure 6.8: Malcolm Baldrige categories
Figure 6.9: Category-Item-Area criteria hierarchy
Figure 6.10: Complete Item-Area hierarchy
Figure 6.11: Who is responsible for addressing this Item?
6.4 Summary

These two examples of the SPRINT prototype system are by no means complete or exhaustive. Their purposes are twofold: (1) to illustrate how the computational representation might be partially realized in a particular programmatic implementation, and (2) to demonstrate how the data collected from actual strategic decision making tasks might be accommodated in a computational model and manipulated through a management support system.

The realization is partial in that the SPRINT prototype does not incorporate some aspects of the representation described in Chapter 5. Currently, relations are not implemented as concepts within the model (i.e., relations are not frames). A relation is a separate object having only two variables: a pointer to another frame (concept), and a weight for the link. Thus, SPRINT only supports binary relations, and default values are not supported for relational weights since a relation object does not include facets for its variables.

The notion of a ‘context’ is implemented in a limited way, mostly due to the fact that relations are not handled flexibly enough to accommodate the requirements for representing multiple decision contexts. A context must include the specific concepts, relations, and properties that are relevant to a particular decision event. If relations were modeled as a type of concept, then a context could simply consist of pointers to the relevant relational and non-relational concepts within the model.
I anticipate that my near-term future research will explore alternative designs for implementing these aspects of the computational representation within the SPRINT prototype for a management support system. The next chapter further explores strengths and limitations of the computational representation in its ability to address the functional architecture for a mental model. This discussion leads to conclusions about research contributions and future research objectives in the last chapter.
CHAPTER 7
DISCUSSION

Earlier chapters have suggested that mental models serve as surrogates for corresponding real-world systems when decision makers attempt to understand the form, function, or state of the world. A decision maker qualitatively simulates a real system in order to produce a description of the system's purpose or of its form; to explain why the system functions the way that it does, or why the system is in the state that it is; or to predict future system states when given assumptions about its properties, or given hypothetical changes to its structure or its properties. The forthcoming discussion considers how well the computational representation for mental models accomplishes these purposes.

To the degree that the computational representation is able to fulfill a mental model's purposes, how well can a management support system, based on this representation, accommodate the complex environment associated with managing long-term strategy? Can a computational model of this strategic management domain produce useful descriptions, explanations, or predictions that augment a manager's decision-making abilities? These questions are discussed in this chapter.
7.1 Achieving the Purposes of a Mental Model

Recall the functional architecture for mental models illustrated in Figure 5.1. The presentation of a computational representation has proceeded from the architecture's center (mental representation and knowledge structure) to the second layer (storage and retrieval processes). This chapter's task is to describe how these structures and processes accomplish a mental model's purposes: To describe, explain, and predict the form, function, or state of an external real-world system.

7.1.1 Description

To describe a concept in a mental model, one would need to include the concept's role in hierarchical structures, its local and inherited properties, and its meaning determined by its relations with other concepts. Such a description is context-dependent in that certain roles, properties, or relations are only relevant in a particular decision-making event, to a particular group of stakeholders, or in a particular time frame. Further, one context may be limited to general or abstract concepts while another is concerned with every available detail. Descriptions must be sensitive to these contextual requirements.

A concept's role is partially determined by its specialization of more general concepts. The primary structural relation in the representation presented here is the IS_A link that relates an instance to its schema or one schema to a more general schema. In the SPRINT prototype system, the IS_A link was given a more intuitive name, 'filedIn',
that corresponds with the filecard/filebox metaphor. In the organizational model, both a Person and an Organization serve the role of an Agent. That is, wherever an Agent can act within the model, a Person or Organization can act as well. If an Agent can perform a Function, a Person can also.

A concept's role may also be described by its associations with other instances of the same schema. In the organizational model, shown in Figure 4.3, two such relations are represented. One 'Function' instance is related to another 'Function' instance via the 'whyPerformed' relation; that is, one business function is a specialization of another, but both are instances of the 'Function' schema. This recursive relation can be carried to any desired level of detail. A function's role is specified within the set of business functions rather than within the set of schemata. Similarly, one 'Organization' is related to another via 'hasParentOrganization', forming a hierarchy among instances.

A concept is also described by listing its properties, both local to that concept and inherited from those more general concepts describing its role in the model. A 'Person' has a 'salary' property, and it inherits properties for 'name' and 'address' from the Person's role as an Agent.

Finally, and perhaps most importantly, a concept is described by its relations with other concepts. The decompositional meaning of a node in a semantic network might be defined by the exhaustive spreading activation to all concepts connected to it (Quillian, 1967, 1968; Collins & Loftus, 1975). An organizational function's meaning is defined by the products that are addressed by that function, the stakeholders of those products, cost drivers that cause the function to be performed, the people who perform it, the
organizations where those people work, the managers of those organizations, and so forth until all relations, and relations of relations, have been exhausted.

Obviously, in a fully-connected network all nodes would eventually be activated. Thus, the spreading activation search algorithm defined in this thesis includes constraint conditions, a decay rate, and a threshold value to limit the decompositional meaning of a concept to a more relevant subset of an exhaustive search. There are no "correct" values for these constraining parameters. Their values depend on the decision context and, like most heuristics, relevant concept nodes may often be screened out along with irrelevant nodes (Cohen & Kjeldsen, 1987; Kjeldsen & Cohen, 1987).

If the model contains multiple decision contexts (i.e., multiple collections of concepts, properties, and relations), then explanations via spreading activation can be delimited by restricting the search to nodes and relations within the context, or contexts, of interest. The use of contexts has not been exploited in the current SPRINT prototype system. Support for contexts, and their use in constraining spreading activation search, will be examined in the near-term future.

7.1.2 Explanation

Explanations are often requested in the form of 'why' or 'who' questions. Why is a particular function costing so much money? Who spends the most time working on functions that address this product? Or, in the TQM strategy, Why did we score poorly on this Baldrige Category? Who is responsible for the activities that address this Item?
When using a model to explain how the corresponding real-world system works, one seeks a sequence of related events, processes, or structures that connect two concepts. These two concepts may represent either temporally separated events (e.g., completed activities in the TQM model) or physically separated structures and processes (e.g. organizational work groups or business functions). Explanations may be produced in a manner similar to that for descriptions except, rather than reporting all relations and concepts surrounding the target node, an explanation seeks one or more paths between two specific nodes or between a node and a specific goal criterion. Each path, as was described in Chapter 5, consists of alternating non-relational and relational concepts.

Although explanation begins with breadth-first spreading activation search, it terminates with a relatively small set of paths through the network that satisfy the original query. Figure 6.6 and Figure 6.11 illustrate the results of explanatory searches in the organizational model and in the TQM strategy model. The SPRINT prototype system displays search output in a reasonably readable manner as a result of the model semantics defined through the fileboxes (schemata). Figure 6.11 explains how Jean Britt-DeYoung is related to a particular Baldrige item. Each concept is further explained by noting its schema, i.e. its role in the model.

An interesting extension of this research would be to translate these search paths into english language sentences. Quillian (1967, 1968) has shown an early attempt to produce such natural language output from spreading activation search in a semantic network. Figure 6.11 might produce the output: "The Malcolm Baldrige item, define planning process, consists of conduct planning and feasibility studies which is an area to
address. This area to address consists of analyze customer requirements which is an area to address. This area to address is addressed by the completed or on-going activity, assess agent satisfaction. This activity addresses the function, build/maintain customer services, which is performed by the organization, cash management services, which is managed by Jean Britt-DeYoung. While this output is not exactly eloquent prose, one could define syntactic rules for translating the current SPRINT explanations into such sentences.

The representation offered in this thesis provides the basic structures and processes for generating such explanations about the real-world system being modeled. The propositional network associates a wide range of concepts about the domain being modeled, while the schema structures describe a concept's role which becomes an integral part of the explanation. The depth and breadth of the resulting explanation depends, in part, on the schematic structure and on how well integrated the network has been constructed (INTEGRATIVE MANDATE), thus underscoring the relevance of a cognitive complexity theory and the importance of symbiotic structures and processes. Search processes are limited by the structure on which they operate.

7.1.3 Prediction

It seems that mental models are uniquely qualified in their abilities to qualitatively predict the futures of complex real-world systems. How well are computational models able to duplicate this capability of mental models? deKleer & Brown (1983) describe a method whereby mechanistic mental models solve problems through propagation: "One
starts with a single, noncausally produced event (e.g., an input or a state of disequilibrium), then examines the nearby components to determine what events resulted, and repeats the procedure indefinitely. This technique has the advantage of automatically constructing the causal relationships between events, and at the same time identifying the mechanisms of the causal relationships” (p. 160). The investigation of such propagated predictions has been very limited in this thesis, but the representation appears to support further work in this area.

Propagation is generally enabled by activated procedures embedded within concepts’ properties in the model. In a frame representation language, the procedures are stored as ifNeeded, ifAdded, ifRemoved, or ifAccessed facets in a concept’s property slots. A limited example of propagation was tested in the Hoshin Management model where a process performance measure (PPM) concept was associated with the strategy that it measured, e.g. ‘Wave solder defect rate’ measures ‘Improve printed circuit board quality’. The strategy would include properties indicating the PPM’s desired value and one or more levels of variance that should trigger managerial corrective action. According to the above description of propagation, one might input a new value for the PPM, ‘Wave solder defect rate’. This input event would trigger an ifAdded demon which causes the PPM concept to notify all strategies that it measures. The strategy would calculate a variance by comparing the new value of its PPM with the strategy’s desired value. The newly computed variance would cause the strategy’s corrective action properties to test whether the user of this model should be notified of the situation.
A similar example of propagation could be developed for the FACT model of an organization. Recall that a 27-rule heuristic was embedded in the ‘recommendation’ property of the Function schema. By adding or removing a person, a function, a cost driver, or a stakeholder comment, or by changing a property in one of these existing concepts, one can trace the propagation of activated procedures in the model to predict the original action’s impact on a function’s recommendation property. These propagated events may be used to predict the effects of actual or hypothetical changes in the model.

7.2 Representational Capability

To what extent is the computational representation able to duplicate the capabilities of the mental representation, so that the computational model might adequately reflect the user’s mental model? This issue has already been partially addressed through matching the representational propositions in Chapter 5 with the theoretical framework and with the case study data. This iterative, triarchic research design was described earlier and will not be further discussed here.

There are, however, two additional aspects of the representational capability that are considered here. First, since no manager works, or thinks, in an isolated world, how can a mental model include beliefs that are attributed to other stakeholders in the strategic environment being modeled? Secondly, how can the representation accommodate ternary and higher-order relations for propositions that cannot be expressed in binary form?
7.2.1 Attributing Beliefs to Others

The propositional networks described in this thesis are solely the expression of one agent's beliefs; one person's or one organization's perspective of the world. Further, it has been assumed that all propositions are positive and are believed to be true, i.e. no assertions were made such that "John Seeley performs Coordinate Corrective Action is false." The simplifying assumptions made thus far reduce complexity of the search algorithm and of the SPRINT user interface. However, it may be important for a computational representation to support models where beliefs of varying truth/falsity are attributed to others.

When someone wishes to represent a proposition as being believed, or not believed, by one or more other agents, he or she might express a nested proposition: S believes (X relation Y) with an associated truth or falsity. That is, James believes that some stakeholder (a person or organization) is interested in a particular strategy, whereas Susan disbelieves the same proposition and a third person suspends judgement on the proposition (neither believes nor disbelieves it). The existing computational formalism can accommodate these extensions in a straight-forward manner.

In this example of nested propositional statements, the 'believes' relationship is just another binary relation that associates an agent, S, with another relational concept that has its own subject and object. Figure 7.1 illustrates this situation. The rectangles in the figure are non-relational concepts, the diamonds are binary relational concepts, the ovals are schemas, and the circles are properties. In order to represent the case where Susan believes that this proposition is false, a 'veracity' property may be added to the
"believes" relation, and the value facet of veracity may be set to False. The veracity property might have a "type" facet restricting its values to True, False, or Nil in the case of suspended judgement.

As an alternative formulation, one might just use the "importance" property of the believes relation, and assign a positive importance when the agent believes the proposition to be True, or a negative importance to propositions believed False. A value of zero could indicate suspended judgement. An added benefit of this alternative representation is that the importance value could range between 0 and 1, or between 0 and -1, to indicate subjective belief, i.e. degrees of belief less that total conviction.
No commitment is made for one or the other of these representations for veracity, but they are the subject of future investigation. In concordance with this thesis' objective, the choice should be made through consideration of a management support system design. An all-or-none form of veracity may be preferable due to its simplicity and intuitiveness for a manager using the system, in spite of the theoretical preference or representational power of subjective beliefs.

Support for multiple perspectives in a computational model is crucial when extending this research into multi-agent reasoning for coordinating strategy. For example, in the Alpha-Tech case study the managers must consider the overlapping, and sometimes conflicting, objectives, strategies, and plans of other functional managers involved in new product introduction. It may be necessary to view the world from another agent's perspective---to qualitatively simulate another manager's position---in order to identify coordinative opportunities and coordinative failures. The mechanisms for this perspective-taking ability are beyond the scope of this thesis, however this discussion has indicated how the computational representation can be extended in this direction. A similar extension is considered in the next section.

7.2.2 Higher-Order Relations

The examples and case study results described in previous chapters are limited to binary relations. While this is not a crippling limitation, it is cause for concern. Lenat & Guha (1990) claim that "a large percentage of the useful predicates in the world have arity 2 or less" (p. 59), because "we believe it mostly just reflects human cognitive
limitations. It reflects the loosely coupled and hierarchical nature of the mental world we think in” (p. 69). This tendency toward binary predicates is balanced by the use of schematic knowledge structures that allow complex concepts to be expressed through collections of simple propositions.

In a similar vein, propositional networks, formed by an interlaced set of binary propositions, support the ability to express one’s understanding of a complex system, such as an organization. The network structure supports other manipulatory procedures like spreading activation search for describing or explaining the complex structure or for testing assumptions about potential relationships between indirectly related concepts. There are, however, limitations to the exclusive use of binary propositions.

To express the fact that Y is between X and Z requires a ternary relation among the three objects. A similar relation is required to represent an agent giving some object to another person. Such higher-order relations are easily accommodated by the representational formalism described in Chapter 5. Relational concepts for ‘between’ and for ‘giving’ would be created as subschemas of a ‘Ternary Relation’ schema. The ‘giving’ schema, for example, might specify typical properties for ‘agent’, ‘object’, and ‘subject’. This schema would be instantiated for each occurrence of the ‘giving’ relation among three instances of non-relational concepts.

However, such higher-order relations create difficulties for the design of a management support system. These concerns are described in the next chapter among the issues for future research.
7.3 Evaluating a Model

A model may be evaluated on either its structure or its content; this distinction was made earlier within the discussion of cognitive complexity theory. A model’s structure is determined by its configuration of hierarchically or flexibly integrated dimensions within and between schemata, whereas the content of a model is only determined when a particular real-world object is placed on one or more dimensions. Placing an object on a dimension might involve simple categorization (INCLUSION) or discriminating finer shades of gray (DISCRIMINATION). For example, in the model of an organization, the structure is described by typical relations such as Function performedBy Person, or by adding a procedure to the Function schema which combines its contribution and reliability to compute its worth. The model’s content is specified when a particular function, e.g. “Determine Corrective Action,” is added as a Function instance or when this instance is specified to have a contribution value of ‘6’. This research is primarily concerned with a model’s structure. The contents of the models for Total Quality Management and for organizational productivity are only used to exemplify their structures.

A model’s structure and content might be evaluated on three criteria: accuracy, consistency, and completeness. The accuracy of a model encompasses both the truthfulness of its contents as well as the precision of values in its properties. The position taken in this research is that a computational model can be designed to represent the user’s beliefs, regardless of their truth or falsity.
A model's consistency might be evaluated both among two or more models held by one person and among the models of different persons. The consistency of structure assesses whether two models have defined the same concepts, properties, and relations. Two models have consistent content to the extent that their common concept instances have equivalent property values and that pairs of concept instances share the same relations with one another. Inconsistent structure and/or content between the models of different managers does not necessarily indicate erroneous models; rather, the inconsistency may highlight a difference of opinion and an opportunity for reducing the equivocality among individuals.

A model's completeness is a subjective criterion that depends on an individual manager's decision making style. These stylistic differences, especially with respect to a model's structure, were described as part of the complexity-theoretic propositions in Chapter 2 (Cognitive Style; Excessive Differentiation). Completeness of a model's content depends on the level of abstraction that is appropriate in a particular decision-making context. For example, a complete definition of an organization's functions, i.e. all instances of the Function concept and the hierarchical relationships among those functions, depends on the decisions that will be based on that model. Corporate-level strategic decisions are associated with more broadly defined functions which may be perceived by more focused individuals as incomplete functional descriptions. A heuristic used at Gamma Financial Services is that an organizational model should have 250-300 functions, regardless of whether the model is for the entire
corporation or for one small department. Through this heuristic, the "appropriate" level of functional detail will be attained.

These evaluative criteria for mental and computational models are necessarily subjective in order to accommodate the differences among individuals' cognitive styles and to reflect the varying demands of different decision-making contexts.

7.4 Modeling Strategic Management Situations

Strategic management situations, like those described in Chapter 4, should first be summarized in a management language which includes the major concepts and relations. A management language of this sort is represented in the linguistic level of the computational architecture. It is not expected that this initial description would be exhaustive, or even completely accurate. A flexible modeling environment, especially as described by the OPENNESS proposition in Chapter 5, should allow such an initial model to be refined and modified to keep pace with the dynamic business environment (FLEXIBLE STRATEGY).

Assuming that one can construct an initial model of the concepts, relations, and properties for the relevant domain, what would a manager do with this description? I appeal to the functional architecture for mental models in responding to this question. The linguistic-level description addresses the mental representation at the heart of the functional architecture. The use of a mental, or of a computational, model cannot be discussed without also considering the model's purposes (description, explanation, or
prediction) and the intervening storage and retrieval processes. These three aspects of a model are symbiotic; one cannot be considered in isolation from the others.

Several processes have already been considered for manipulating a model's representation. Spreading activation was described as a means for producing descriptions and explanations by traversing the relations in a representation. The schema-instance hierarchy supports inference through structural inheritance. Activated procedures embedded in a frame's slots support procedural inference techniques for deriving the values of particular properties and for predicting the outcomes of changes to the model. Through examples drawn from the management of a TQM strategy and from the use of an organizational model to improve productivity, it was shown how a computational model might answer managerial questions in these two domains.

A manager's ability to use a computational model to augment his or her decision making processes is directly related to the ease with which the model can be created and accessed. A complete representation, alone, cannot assure a successful management support system. Nor can a fully functional user interface in a management support system assure that the resulting system augments strategic management. Just as representation and processing are symbiotic in a computational model, so are the model and user interface symbiotic in designing a successful management support system.

The design and evaluation of a management support system user interface are beyond this thesis' scope and were described in Chapter 6 only to the extent required to illustrate how a system could be implemented for manipulating a computational model.
The future research topics outlined in the next chapter include both concerns with the current SPRINT prototype system and suggestions for improving it.

7.5 Using SPRINT in Practice

Stated generally, the SPRINT management support system should be useful in tasks where multiple dimensions are associated via a relatively stable semantics in order to assist managers in remembering associations and in synthesizing relevant dimensions when identifying problems and opportunities. This research is limited to demonstrating how the prototype system would be applied to problems such as those described in Chapter 4. No attempt is made to assess how well managers are in fact able to work with such a system.

Tentative conclusions may, however, be drawn from the interviews conducted during the two case studies. It seems likely that a knowledge engineer's assistance would be required for most managers attempting to develop model semantics such as those shown in the figures in Chapter 4. It is not suggested, or even recommended, that these semantics reflect how an "expert" would describe the corresponding real-world system. Rather, these semantics should reflect the language actually used by the manager employing this model. Although, there does appear to be an opportunity for using models such as those developed for the TQM strategy and for the organizational productivity FACT analysis method in order to teach such "expert" models to other managers (FLEXIBLE LEARNING).
Once a schematic knowledge structure is defined and is implemented in SPRINT through use of the filebox metaphor, the user is responsible for all dynamic changes to this model. The principle changes are addition of new filecards to existing fileboxes and creation of links between pairs of filecards. Both of these changes are easily accomplished through the existing SPRINT interface. It is unlikely, however, that most managers would be either willing or able to add or modify filebox definitions, i.e. to change the model structure. This concern must be investigated through future empirical research.
CHAPTER 8
CONTRIBUTIONS AND FUTURE RESEARCH

8.1 Research Contributions

The work presented in this thesis adopts a focus that is decidedly different from decision support systems, regardless of whether the decision support emphasizes individuals, groups, or organizations (George, 1991). My use of the description "management support systems" intentionally avoids focusing on decisions and, instead, emphasizes the need to support management processes as they relate to deliberation and issue identification in strategic situations. The strategic management process is more synthetic than analytic (Mintzberg, 1978). Cognitive complexity theory illuminates the structure and process of this knowledge synthesis, while Image Theory suggests how this elaborated knowledge base would be used in framing decisions.

It therefore seems appropriate to adopt an approach whereby these strategic situations are understood as a manager's mental models of, for example, a strategic planning process, a particular strategy for TQM, or a multifaceted organization's
productivity. After reviewing theories of mental representation, a proposal was made for computational models that would reflect these mental models and for a computational representation that mimics its mental counterpart. The computational model may serve as the core of a management support system for these synthetic cognitive tasks. Through examples drawn from two case studies, it is suggested that such a management support system would augment a manager's own memory and reasoning processes for creating and using mental models.

This research contributes to several theoretical concepts, as well as to the practical application of these research results in the form of support systems for managers in industry. The theoretical contributions:

1. Clarified the definition and role of mental models in strategic management by integrating theories for the structure of knowledge and for decision-making processes that operate on those structures;

2. Defined a computational representation and a formalism which accommodates the theories and which allows computational models to be constructed that reflect the structure of managers' mental models; and

3. Built a computational model using data from the case studies that demonstrates the representation.

The practical contributions may be summarized as follows:

1. Implemented a prototype management support system using the Smalltalk/V286 programming language that provides general facilities for building computational models;

2. Demonstrated how the computational representation in SPRINT serves as an "extended memory" for managers, i.e. a computer-based memory and associated storage and retrieval processes that match the knowledge structure used by a particular manager; and
3. Showed how a computational model may be used for managing strategy implementation such as assigning, monitoring, and evaluating activities which address the Malcolm Baldrige application areas for a TQM strategy.

Several extensions to this research are outlined for future investigation.

8.2 Extending the SPRINT Interface

Using the hypermedia concept for designing a system interface with nonlinear text/media conforms very well to the semantic network structure described in this representation for computational models. By adopting a filebox/filecard metaphor in the system design, the basic hypermedia concept is extended to accommodate the schematic knowledge structure. However, this system interface design imposes its own limitations on the user's ability to access the underlying model representation.

Recall that a proposition is expressed as a binary relationship between two concepts and is represented as a relational concept that serves as an intermediary between two non-relational concepts. A binary relational concept has two properties that hold the subject and object of that relation. In a hypermedia interface, the intermediary relational concept is transparent to the system user who sees the relation simply as a named link with which one can traverse the network from the current node to another concept node. This neat correspondence between the representation and hypermedia breaks down when extending the knowledge structure beyond binary relations or beyond a single-perspective propositional network.
The complications described in the last chapter---attributing beliefs to others and expressing higher-order relations---have been avoided in the present research due to the difficulty of accommodating these representations with a hypertext interface. Consider the belief attribution shown in Figure 7.1. If one were to traverse the 'believes' link from Susan Jeffries, how would the destination node be presented? The hypermedia interface generally works on the premise that the user moves from one concept node to another, not from a concept to a relation. While this problem doesn’t seem to be insurmountable, its resolution is postponed to future research.

An analogous concern and postponement applies to including higher-order relations in the model. The hypertext interface makes binary relations transparent to the user simply by passing through the relation to its object property. No such transparency can be applied to higher-order relations since two or more choices always exist when traversing the relation.

An additional topic for further investigation is to improve the graphical display of descriptions and explanations derived from spreading activation search. In a limited way, SPRINT's graphical display browser, described in Chapter 6, can present concept descriptions by displaying one relational dimension at a time. For example, showing a Function performedBy a set of Persons. This display could be expanded to show several relations at once, with the lines corresponding to each relation-type assigned a different color. A Function could be displayed with performedBy (Person) relations shown in red and causedBy (Cost Driver) relations shown in green. Thus, a more complete description for the function's decompositional meaning is presented to the system user.
A final consideration for enhancing the user's interaction with SPRINT is to develop a grammar and language for creating, modifying, or traversing the model. This grammar would be based on the representational formalism presented in Chapter 5. A manager interacting with a SPRINT model through its graphical interface would not, and should not, have a direct need for such a grammar and manipulation language. This language would be used by advanced users for creating activated procedures embedded in the properties' facets and for specifying goal and constraint parameters for the spreading activation search. The grammar and language would also facilitate development of graphical user interfaces by providing the programmer with a language for manipulating the knowledge base.

### 8.3 Mental Models and Coordination

It appears that there are at least two reasons for pursuing research into multiple interacting models: (1) an individual person often uses several models when making a single decision; and (2) multi-agent reasoning at the organizational level involves interaction among several managers, each using his or her own model. My future work in this research stream will emphasize the latter problem as a means to design an organizational support system which assists several managers (each using an individual management support system) to coordinate their strategies.
Organizational-level coordination is well illustrated by a quote from a manufacturing engineering manager in the Alpha-Tech case study. He described the early stages of issue deliberation and project management as follows:

I'll start with my list [of actual and potential projects] and then maybe open it up to a broader group within my area, or pull in people from Manufacturing Specs or so forth, and just do some brainstorming. What's the potential impact on us? Get their inputs, then sit back and look at the inputs and say: Are they valid or are they non-issues? From there you start allocating, well, how am I going to get resources to do this thing? That's where you start the prioritization scheme of swapping. You put things on the back burner; you pull things off. You just try to get the time you feel is necessary to meet the need. And play it from there. ... That's the real challenge: Don't be the bottleneck!

In things like that, of planning out the schedule, you have to have a good feel for: What's our role in the project? What part do we play in it? Then, take that into these project team meetings to say: These are what we see as being our responsibilities. How do they mesh with everyone else, what your needs are of us, and what our needs are of you? It's a two-way street. A lot of it is the definition of what your responsibilities are. What are we as Manufacturing Software Engineering responsible for? You get back down to your mission statement. Why do we exist? You kind of bind it around that. That's where you take your cut.

In a study conducted by Ancona & Caldwell (1990), the product development process is divided into three phases: creation, development, and diffusion. They state that most team interactions during the creation phase fall into one of three categories: collecting information or resources, modeling the organizational environment, and building links with other groups. In addition to collecting information, team members attempted to create models of how other groups would respond to the product. Ancona & Caldwell conclude with a recommendation that, "Clearly, applications need to be developed to encourage teams to model their dependence on the external organization and
to develop ways to meet that dependence." My future research is directed at applying the mental model representation in a management support system for assisting a manager in coordinating his or her activities with those of other managers.

An interesting coordination framework for agents is outlined in Gasser, Rouquette, Hill, & Lieb (1989). An organization is described as "a particular set of settled and unsettled questions about belief and action through which agents view other agents." Thus, an organization is defined by "interlocking webs of commitment" rather than by structural relationships among a collection of agents. Organizational change, then, means "opening and/or settling some different set of these questions in a different way, giving individual agents ... a different context of assumptions about the beliefs and actions of other agents." An agent's action choices are constrained "because of what it believes other agents expect it to do (its commitments to others), what it expects itself to do (its commitment to its own future actions), and because of what it expects of others (its beliefs about the commitments of others)." Collectively, agents' expectations or default assumptions for actions of other agents are the organization.

Another description of how individuals may coordinate their actions in an organization is given by Weick & Bougon (1986): "Concerted action is possible where there is common relevance of two concepts in two cause maps and a double interact to link the maps" (p. 110). Two concepts may become linked by a double interact if: Two actors, A and B, both with quite different cause maps, become organized when person A wants X and person B has the capacity to fulfill X, and when person B wants Y and person A has the capacity to fulfill Y. Even though concept X is an element in the cause
maps of both A and B, the meaning of that element differs in each map because it is connected to different ideas. Weick & Bougon note that "This process is not easy, since it requires that one person reconstruct someone else's definition of who he or she is." That is, if person A is to construct a double interact for coordinated action, he must understand B's wants and beliefs.

The above description of organizations as interlocking webs of commitment or as intentionally constructed double interacts may appear too rationalistic. Some theorists view organizations as being more akin to a temporary moment of order in a world of chaos. An open system theory characterizes organizations as consisting of a loosely linked coalition of relatively autonomous members whose outcomes are strongly influenced by environmental factors and by each other. Further, environmental information arrives to organizational members asynchronously, possibly with contradictory interpretations (Scott, 1987). Open system theories often assume a rational model (albeit, limited or bounded rationality) where organizational members pursue relatively specific goals and operate in a relatively formalized structure (March & Simon, 1958). An alternative approach is to consider organizations as open, but natural systems where behavior of members is seen as more important than the normative structure of rational theories. Natural system theorists view organizations as collectivities whose members share a common interest in the survival of the organization, but with less emphasis on the pursuit of specific goals (Scott, 1987; March & Olsen, 1976).

My initial future research will adopt an open system framework for modeling coordination among relatively autonomous agents where some degree of rationality is
imposed on the structure of individual strategic plans. In particular, the notion of an "interlocking web of commitment" from Gasser, et al. will be investigated for its presence in each member's mental model of the strategic management situation. This web of commitments will be represented in a network of corresponding computational models. This open system model for coordination also contributes to a general architecture for designing distributed information systems.

8.4 Distributed Knowledge-Based Systems

We have previously established a research stream investigating an architecture for integrating distributed knowledge-based systems (Carlson & Ram, 1989, 1990a, 1990b, in press). This thesis' contributions help advance our goals for continuing research into distributed systems for supporting strategic planning. Our prior work in this area described how messages would be exchanged among distributed knowledge-based systems (KBSs), with the interactions controlled by a centralized distributed knowledge-based management system (DKBMS). Each KBS would represent one semi-autonomous agent in a distributed network, i.e. an organizational unit or an individual manager. A global knowledge-base in the DKBMS would analyze the terms of a message or query through meta-knowledge about the planning domain and through a semantic network containing common planning terms. The DKBMS forwards messages to what it determines to be the most relevant KBSs, and it serves as a dialogue manager for tracking on-going exchanges.
It is anticipated that the dialogue manager could monitor exchanges among KBSs according to a speech-act theory conversation structure (Winograd & Flores, 1987; Winograd, 1988). The entire set of speech acts may be classified into five categories (Searle, 1975):

**Assertive:** Commit the speaker to the truth of a proposition

**Directive:** Attempt to get the hearer to do something (questions and commands)

**Commissive:** Commit the speaker to some future action (a promise)

**Declarative:** State the correspondence between a proposition and reality (pronounce a couple married)

**Expressive:** Express a psychological state (apologize or praise)

---

**Figure 8.1:** A basic conversation for action
The dialogue manager might track the progress of a "conversation" between two KBSs according to a state diagram of logically possible exchanges. The diagram shown in Figure 8.1 is reproduced from Winograd & Flores (1987, p. 65). At each point in the dialogue, there is a small set of possible actions determined by the previous history. The heavy circles shown in the figure signify "completion" states from which no further action can be taken. We expect to use a similar state diagram for developing a dialogue manager within the DKBMS.

The global knowledge base in the DKBMS might also contain domain-specific rules or procedures for identifying potential inconsistencies between two or more individual KBSs. For example, two interdependent plans might not be synchronized, or the predicted outcome of one functional manager's plan may not fulfill the goal of the general manager's strategy for the business unit. Such multi-level coordination would be beneficial in the Hoshin Management Process used at Alpha-Tech Corporation.

The computational representation described in this thesis suggests one design for an individual KBS in this distributed architecture. By providing a SPRINT system to each of several managers, and by linking those systems via a DKBMS which understands, for example, the Hoshin Management Process, the distributed SPRINT network might assist managers in coordinating their respective roles in the organization's strategies. The Alpha-Tech case study may be extended to support this continuing research stream.
APPENDIX A: INTERVIEW PROTOCOL

1. Demographics.
   a. Name, Title, Organization chart, Time in this job, Time in this company?

2. Structure of a plan.
   a. What is the mission of your department within this company?
   b. What are some of the major issues facing your department?
   c. Are any of these issues shared by [name other departments in this study]?
   d. Describe your role in the most widely shared issue?
   e. What are the causes of this issue?
   f. What do you plan to do about this issue?
   g. How will you know if you do or don’t succeed?

3. Mental models of others.
   a. What other persons or organizations are important for your plan?
   b. What role do these other persons and organizations play in this plan?
   c. How is responsibility separated among these people?
   d. When you decide whether or not to inform someone else about a change in your plans, what kinds of things do you consider?
   e. Do you consider what the other person will do with the information that you give to him/her?

4. Social worlds.
   a. What if you don’t know that other person very well?
   b. What are some of the ways that you “group” people into categories?
   c. Describe the assumptions that you make about [pick a group mentioned by R]?

5. Communication and Coordination strategies.
   a. How do you keep your plans and activities coordinated with those other managers?
   b. How do you check the progress on the plans of others which are required for successful completion of your plan?
   c. How do you decide when to communicate status information on your plan to other managers? What do you tell them?

6. Access to information.
   a. How do you keep track of this information now (files, notes, spreadsheets, etc.)?
   b. What other information would you like to have that you don’t have now?
   c. Can you think of any ways that a computer system might help you with this information?
APPENDIX B: NATIONAL QUALITY AWARD CRITERIA

The Malcolm Baldrige National Quality Award was signed into law on August 20, 1987. The annual award recognizes the achievements of those companies which improve the quality of their goods and services and which provide an example to others. The award application criteria specify (in 1990): 7 Categories, 33 Items, and 133 Areas to Address which permit evaluation of an organization's quality program. The categories and items are summarized below.

1 Leadership
   1.1 Senior Executive Leadership
   1.2 Quality Values
   1.3 Management for Quality
   1.4 Public Responsibility

2 Information and Analysis
   2.1 Scope and Management of Quality Data and Information
   2.2 Analysis of Quality Data and Information

3 Strategic Quality Planning
   3.1 Strategic Quality Planning Process
   3.2 Quality Leadership Indicators in Planning
   3.3 Quality Priorities

4 Human Resource Utilization
   4.1 Human Resource Management
   4.2 Employee Involvement
   4.3 Quality Education and Training
   4.4 Employee Recognition and Performance Measurement
   4.5 Employee Well-Being and Morale

5 Quality Assurance of Products and Services
   5.1 Design and Introduction of Quality Products and Services
   5.2 Process and Quality Control
   5.3 Continuous Improvement of Processes, Products, and Services
   5.4 Quality Assessment
   5.5 Documentation
   5.6 Quality Assurance, Quality Assessment and Quality Improvement of Support Services and Business Processes
   5.7 Quality Assurance, Quality Assessment and Quality Improvement of Suppliers
6 Quality Results
6.1 Quality of Products and Services
6.2 Comparison of Quality Results
6.3 Business Process, Operational and Support Service Quality Improvement
6.4 Supplier Quality Improvement

7 Customer Satisfaction
7.1 Knowledge of Customer Requirements and Expectations
7.2 Customer Relationship Management
7.3 Customer Service Standards
7.4 Commitment to Customers
7.5 Complaint Resolution for Quality Improvement
7.6 Customer Satisfaction Determination
7.7 Customer Satisfaction Results
7.8 Customer Satisfaction Comparison
APPENDIX C: DIMENSIONS IN ORGANIZATIONAL MODEL

Contribution (Value-Added contribution to the organization):

10   Adds Maximum Value -- Adds considerable value by increasing revenues and/or creates market leadership in customer service and quality. Relates directly to the primary mission and core business. Strong desire to expand such activities.

8    Adds Significant Value -- Very important, adds value and enhances competitive position.

6    Adds Limited Value -- Maintains competitive position by slightly improving the delivery system.

4    Maintains Existing Value -- Maintains the status quo. Included are activities required by law.

2    Marginal Value -- Adds little or no value to future delivery of products and services (e.g., problem solving activities). Internally required, but adds little external value. Strong desire to reduce such activities.

0    Non-productive -- Contributes no value and should be eliminated if possible.

Reliability (as perceived by the customer):

10   Perfect -- rarely used. The function is performed in every aspect without error.

8    Performed well with infrequent, or rare, errors where the faults cannot be identified by the customers or users of the products or services.

6    Performed with few problems in which the customer or user may not be able to quantify. The problems tend to change over time and therefore do not represent chronic problems associated with the delivery of products or services. Problems tend to be related to a failure to adhere to a proven or reliable system.

4    Known problems exist in the delivery system and the manner in which products and services are delivered.

2    Unreliable, implying that the organization cannot produce the desired results in any reliable fashion required to meet customer requirements.

0    Terrible.
REFERENCES


