EXPLORING AGENT-BASED SIMULATION OF CAUSAL MAPS: TOWARD A STRATEGIC DECISION SUPPORT TOOL

A dissertation submitted to the
Kent State University Graduate School of Management
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

By

Douglas Allen Druckenmiller

May, 2005

Dissertation written by

Douglas Allen Druckenmiller

B.A., Philosophy, University of Illinois, 1972

M.M., University of Mary, 1990

Ph.D., Kent State University, 2005

Approved by

_____ Co-Chairman, Doctoral Dissertation Committee
William Acar

_____ Co-Chairman, Doctoral Dissertation Committee
Marvin D. Troutt

_____ Member, Doctoral Dissertation committee
James W. Boyd

Accepted by

_____ Doctoral Director, Graduate School of Management
Donald R. Williams

_____ Dean, Graduate School of Management
George E. Stevens

TABLE OF CONTENTS

Chapter 1. Introduction 1
1.1 Business Intelligence Systems 3
1.2 Research Question 7
1.3 Research Method and Structure of the Dissertation 7
Chapter 2. Background 9
2.1 Information Requirements 9
2.2 Processing Requirements 18
2.3 Technical Requirements 28
Chapter 3. An Agent Architecture for Comprehensive Situational Mapping (CSM) 38
3.1 CSM Dialectial Inquiry 38
3.2 CSM Semantics 39
3.3 CSM Process 41
3.4 Levels of Analysis 42
3.5 Proposed Agent Architecture 44
Chapter 4. Prototype Design 49
4.1 The Problem of Automated Support for Causal Mapping 49
4.2 Prototype Development Platforms 50
4.3 User Interface Design 52
4.4 Agent-based Simulation of Causal Maps 58
Chapter 5. Prototype Testing 69
5.1 Semantics Testing 69
5.2 Process Testing 71
5.3 Usability Testing 76
Chapter 6. Conclusion 92
6.1 Research Contribution 92
6.2 Future Research 93

APPENDICES

Appendix A: Usability Test Documents 95
A.1. Introduction (read to all testers) 95
A.2. Task Scenario – ABC Company Sales (Acar, 1983) 95
Appendix B: Test Tasks 98
B.1. TASKS - Manual System 98
B.2. TASKS - Computer System 99
B.3. TASKS - Additional Computerized 101
Appendix C: CSMtools User Guide 102
Appendix D: Post Testing Survey 108

FIGURES

Figure 1. Problem type and time horizon 4
Figure 2. Types of information systems support 5
Figure 3. Porter's value chain model (Porter, 1985) 11
Figure 4. The ICA's strategic planning process 23
Figure 5. Gap analysis and the formation of strategy 24
Figure 6. Eden's strategy making process 25
Figure 7. CSM bridge. (Acar, 1987) 39
Figure 8. An example of CSM (Acar, 1983) 40
Figure 9. Link coefficient notation - mathematical basis 41
Figure 10. A synoptic view of Acar's system: components of causal mapping. (Acar, 1983) 43
Figure 11. Primary agent roles in the proposed operationalization of Acar's (1983) CSM 46
Figure 12. User interface components 53
Figure 13. Node property sheet. 55
Figure 14. Link property sheet. 56
Figure 15. Simulation controls. 57
Figure 16. Editing interface 59
Figure 17. Repast simulation of edited map – simulation time 170 weeks 61
Figure 18. Repast simulation of edited map – simulation time 177 weeks 62
Figure 19. Repast simulation of edited map – node graph 63
Figure 20. Revised example according to economic theory. 64
Figure 21. Economist scenario at end of simulation. 65
Figure 22. Test map of infinite causal loop. 67
Figure 23. Jay's map 70
Figure 24. Vivian's map 71
Figure 25. *Scenario A* simulation results for Jay's map 72
Figure 26. *Scenario A* graphed goal nodes for Jay's map. 73
Figure 27. *Scenario A* simulation results for Vivian's map. 74
Figure 28. *Scenario A* graphed goal nodes for Vivian's map. Time scale is in weeks. 75
Figure 29. Test map for partial and restriction link types 76

TABLES

Table 1. Simulation of scenario through first 5 time ticks. 60
Table 2. Table of average timings in minutes and seconds. n=15 86
Table 3. Seven point scale: comparison between manual and computer systems use. 90
Table 4. Five point scale: use of tool for strategic planning. 90

CHAPTER 1 INTRODUCTION

The emergence of the electronic marketplace and the pervasive use of information technology **(IT)** in business have fundamentally changed the way firms operate in the 21st century. Information systems **(IS)** are no longer just a part of back-office support systems but are taking on a larger role in the primary strategic thrust of every firm (Porter, 2001; Tapscott, 2002). The pervasiveness of IT, and the collection, storage and dissemination of information from all aspects of firm operations, overwhelm the strategic decision maker's capacity for formulating strategic problems and developing business strategy. Strategic decision-making in firms remains more art than science (Hoopes, Madsen & Walker, 2003) leading many to propose that managers faced with complex dynamic electronic market environments and systems increasingly depend on intuition and "strategic thinking" to cope with rapid change. For firms heavily invested in IT, every aspect of the value chain, both internal and external, is becoming integrated and dependent on IT save one: support of strategic decision making. While a decision maker's intuitive thinking is a necessary part of strategy formulation, intuition alone is not always reliable. Situations requiring strategic thinking take place in real time. The research problem addressed by this dissertation is the development and testing of a tool for strategic decision support that can integrate with today's electronic market environments.

This dissertation develops and tests a prototype system for strategic decision support that is based in current strategic management theory and incorporates recent developments in distributed artificial intelligence **(DAI)**. The key innovation in the research is the incorporation of agent-based modeling and simulation techniques into the innovative yet integrative causal mapping and dialectical system for scenario planning developed by Acar (1983). Until this dissertation, no working prototype has incorporated an understanding of strategy as contained in the literature, integrating the approach to problem formulation and strategy development proposed by systems thinkers (such as: Ackoff, 1978; Churchman, 1971; 1979; Acar, 1983; Mason & Mitroff, 1981; Mason & Mitroff, 1981; Acar, 1983) with the latest distributed artificial intelligence technology. This dissertation discusses the underlying strategic theory and the requirements of a knowledge-based strategic decision support system and derives system requirements. The basic problem of representing strategic knowledge is addressed and an agent-

based modeling approach to causal mapping is developed. The tested prototype reveals the directions for further research and development of the system as a viable tool for strategic decision support.

Simulation of strategic environments using complex causal maps containing multiple feedback loops presents an analytically intractable problem. This is a common feature of systems modeling where closed-form solutions to modeled problems are not available or are exceedingly difficult because of the non-linearity introduced by the interactions in the model. Simulation techniques allow the models to be explored in order to gain insight into the emergent systems dynamics present in modeled situations. This dissertation will show that this problem can be efficiently solved using agent-based modeling **(ABM)** and simulation techniques. Two basic research questions are pursued through ABM simulation: **i.)** empirical evaluation of system dynamics and **ii.)** development and design of new systems (Daniels, 1999). Both of these questions are relevant to scenario planning and strategic decision support. The interaction of autonomous entities in a common environment is typically a mutually recursive process that is analytically intractable. Causal maps are models of such environments. Simulation of complex systems allows evaluation and observation of global behavior and system dynamics that cannot be analytically predicted *ex ante.* This is particularly true of causal maps containing multiple feedback loops. This dissertation explores creative agent-based solutions for the simulation of complex causal maps.

Strategy is ultimately concerned with the design of complex systems. ABMs provide unexpected insights into holistic patterns based on the dynamic interactions of simple components. Similarly, simulation assists the design of new complex systems composed of multiple interacting agents. This research has great potential as a long-term research project in developing a distributed group decision-support system **(GDSS)** for strategic knowledge management for improved strategic performance and tactical productivity.

This dissertation discusses an approach to support Comprehensive Situational Mapping **(CSM)**, a causal mapping approach to scenario planning proposed by Acar (1983), with agent-based modeling. ABM simulations are multi-agent systems **(MAS)**, a DAI technique. CSM addresses the dichotomy between classic management theory approaches to problem solving. These manifest themselves between the quantitative modeling of Operations Management **(OM)** and Operations Research **(OR)** and pure process consulting, which takes a more intuitive behavioral approach to the process of strategy formulation. This basic dichotomy is reflected in

information systems support, with decision support systems **(DSS)** closely tied to OM and OR modeling support and GDSS supporting process consulting. Fortunately, CSM is a set of structured methods that combines the rigor of OM/OR modeling approaches with process consulting techniques in supporting situational analysis and problem formulation.

1.1 Business Intelligence Systems

Information systems have been classically categorized by the type of decision making that such systems support (Ansoff, 1965; Gorry, Anthony & Morton, 1971). In this classical view, a firm's information system is a business subsystem that operates as a feedback loop and provides the information necessary to identify problems, develop effective solutions, take corrective actions, and control firm operations. The type of management decision making and corresponding type of information support system form a layered management triangle. Firm operations form the base of the triangle and successive layers of management form a management hierarchy, each with its particular decision focus and corresponding type of information support system. Day-to-day firm operations at the base are supported by transaction processing systems that capture the foundational data that other layers use as a foundation. The next layer up from base operations is the operations management layer that works with standardized exception reports which can be used to control operations. Above operational management is middle management which is concerned with the broader picture and operates at a tactical level of control and has a time frame that spans the annual budget cycle. At the next level up in the hierarchy, top management is charged with strategic decisions with a longer time frame and deal with the uncertainties of the firm and its interactions with the environment (See Figure 1).

This triangle represents the management decision hierarchy, the type of decisions that are made at each level, the unique problem formulation that is required at each level and the type of information support that is needed for the formulation of solutions. This is not a suggestion that managers should necessarily be structured into hierarchies or that the model is inappropriate for matrix organizations, etc. Research (Bresnahan, Brynjolfsson & Hitt, 2002) in fact suggests that companies heavily invested in IT have flat management hierarchies composed of many autonomous business units that perform all roles of the decision hierarchy. The decision hierarchy is a view of decision making roles that exist in any organization, whether it is a single individual wearing many hats, self-directed teams or other types of management structures that exist in an organization (O'Reilly III & Pfeffer, 2000).

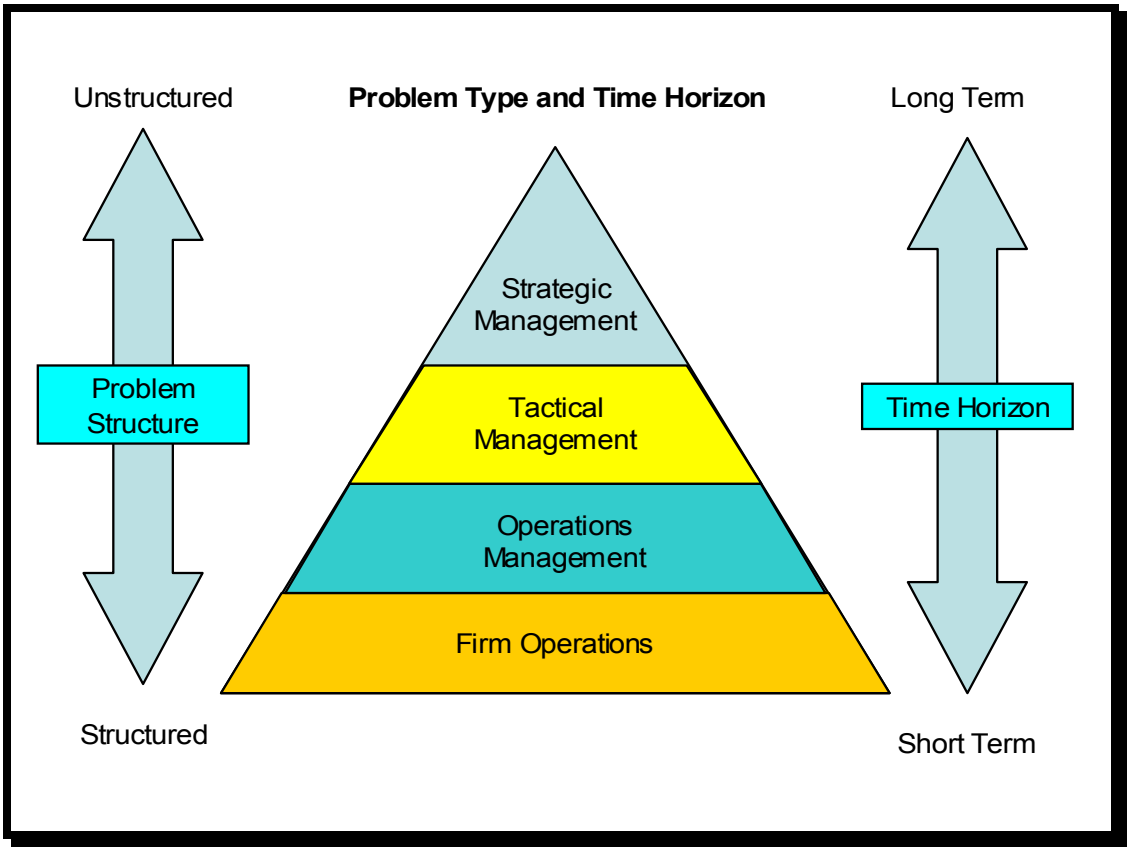


Figure 1. Problem type and time horizon

Information flows upward from transactions at the bottom layers in ever more summarized form and is the basis of decisions flowing downward with increasing specificity. Problems at the base are highly structured and amenable to programmed solutions while problems at the top are highly unstructured and difficult or impossible to specify. This view characterizes the systems approach to problem solving and, in the past, formed the foundation of management information systems design and implementation.

Looking at Figure 2, older automated data processing systems **(ADP)** and newer database management systems technology **(DBMS)** have totally automated the base of the triangle for even the smallest business. Classic management information systems provide routine reporting to managers on operations. Sophisticated DSS provides analytical processing power to solve the semi-structured problems of middle level tactical management. More recent support for strategic decision making at the top of the hierarchy is manifest through a number of systems designed to

provide strategic management with strategic decision support **(SDSS)**: enterprise resource planning **(ERP)** systems, executive information systems **(EIS)**, expert systems **(ES)**, knowledge management systems **(KMS)**, and GDSS all provide support for some aspect of strategic decision making (O'Brien, 2002).

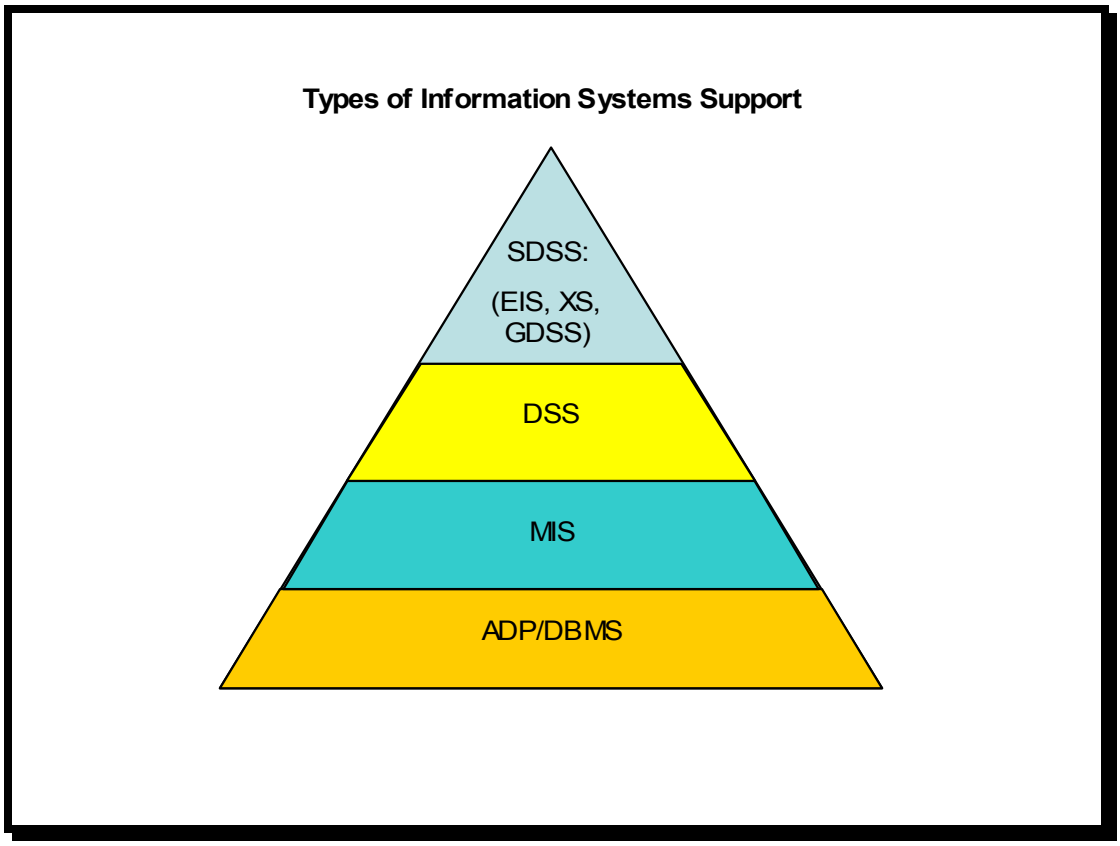


Figure 2. Types of information systems support

Yet these systems by themselves have little to do with formulating business strategy. They may provide knowledge or information that is used as an input to the process and allow sophisticated drill down to lower level information housed in data warehouses or data marts, yet they provide little or no support of the process of strategic thinking and decision making. The design of an "inquiring system" for strategic management needs to take into account the nature of business strategy and the process of strategy formulation as it involves a team of decision makers in an organization and the emergence of inter-networking technologies and electronic markets that provide an information rich but analysis poor environment for strategic management (O'Brien, 2002).

The information systems literature in this area typically takes a different approach to the problem, that is, no regard for what strategy is as defined in strategy research. Recent research (Forgionne & Kohli, 2001) serves as an illustration. Researchers designed an experiment to compare the effectiveness of a management support system **(MSS)** for strategic decision making compared with a traditional DSS system. Their MSS was a traditional DSS system combined with an EIS and ES. They used a management simulation game with two different groups, one using a DSS and the other an MSS. The team using the MSS achieved greater profitability than the team using DSS in the simulation.

The problem presented to decision makers was a "complex, semi-structured management problem." Decision makers had to set the level for four decision control variables (price, marketing, R&D and plant expansion) and compete with other companies for a four-quarter total market potential. The problem is that the time frame (four quarters) and the decision situation (semi-structured) are not really what contemporary strategic literature is dealing with when talking about strategy (Porter, 1996; Phelan, 1997). The control variables and objectives are already well defined. The software supports determination of optimal solutions and decisions. However, this is not the situation that a business strategist confronts. The management problem has no structure and encompasses much longer time frames with a great deal of uncertainty. In addition the management team itself has contradictory explanations and expectations of the strategic situation and future scenarios (Phelan, 1997).

Strategic decision support systems like ES, EIS, KMS and ERP may provide knowledge or information that is used as an input to the process and allow sophisticated drill down to lower level information, but they provide little or no support of the process of strategic thinking and decision making. Such systems are either designed to provide optimized solutions to well formulated problems or information inputs for problem finding. Thus the problem situation has already been framed or is semi-structured and is therefore a tactical problem. Alternatively decision-makers are flooded with information that may not be relevant to understanding and structuring the complex nature and relationships of the strategic situation, thus creating "management misinformation" (Ackoff, 1967).

This dissertation proposes a fresh approach to the issue of strategic decision support. A structured method for problem formulation, Comprehensive Situational Mapping **(CSM)** (Acar, 1983) is used as the basis of a systematic approach to problem structuring. CSM makes explicit multiple decision-makers' cognitive understanding of the problem in an integrated causal

semantic network or causal map. CSM provides a basis for a structured process approach to strategy formulation that integrates the quantitative modeling approach of operations research and intuitive dialectical process consulting. CSM's structured approach to strategy formulation provides the processing requirements for a SDSS and a knowledge representation framework for a GDSS that incorporates knowledge management techniques that can be supported using MAS technology.

1.2 Research Question

A review of business intelligence systems reveals that strategic decisions are characterized by the unstructured nature of the problem and the uncertainty and turbulence of the environment. Current information systems architectures provide overwhelming data and analysis capabilities. The tools are tactical in nature, however. Their focus is on providing optimal solutions to well structured or semi-structured problems appropriate for implementation but not conceptualization of strategic problems and design of strategic alternatives. The current systems still create information overload. There is therefore a need for a strategic control system that insures the delivery of relevant information.

The research question addressed by this dissertation is "How best support the process of strategic decision making in an IT intensive environment?" This question contains two sub-questions which must be addressed:

- What are the content and process of strategic decision making?
- What type of systems architecture will best support strategic decision making?

The order of these questions is important. The first question identifies the system requirements and the second a proposed system design to meet those requirements.

1.3 Research Method and Structure of the Dissertation

The research method used is a standard approach to systems analysis and design followed by the creation of an initial system prototype. Systems analysis determines the basic requirements for the system and systems design determines the overall architecture that provides for a phased development and prototyping of the system. The developed prototype is then based on a central system component that illustrates the feasibility of a phased modular development,

serves as a practical demonstration of a specific technical approach, and provides a focus for further development and implementation.

Chapter 2 details the system requirements through analysis of strategic decision making. A review of the strategic management literature provides the basis for determining both the information and processing requirements of the system. The technical requirements are developed from the published literature on distributed artificial intelligence, which is applicable to the problem domain. Chapter 3 constructs a system architecture based on CSM which provides a multi-agent approach to the process of situation formulation and strategy development. An agent architecture for CSM and supported analytical processes is then developed. Chapter 4 develops a specific prototype system for the creation of causal maps and their forward analyses through agent-based simulation. This chapter provides a simple example of a causal map and its simulation using agent-based modeling technology. The specifics of the prototype implementation and associated issues are discussed. Chapter 5 covers the testing of the developed prototype in terms of its semantic validity, process capability, and functional usability. Chapter 6 concludes the work with a discussion of the research contribution and future research.

CHAPTER 2 BACKGROUND

This chapter looks at the primary requirements for a strategic decision support system from three perspectives: *Information requirements, processing requirements*, and *technical requirements*. An analysis of these requirements provides the basis for the design of a support system that is consistent with strategic management theory and a basis for a comprehensive and integrated solution to the problem. Without a comprehensive approach to systems requirements definition, suboptimal solutions are likely. *Information requirements* come from an analysis and understanding of strategic theory focusing on the **content** of strategy formulation. The debate in the literature is between a focus on an analysis of competitive forces and industry factors versus the internal resources and capabilities of the firm, and the activity fit between external factors and internal capabilities. *Processing requirements* are also found in the strategic management literature in which there is a focus on the **process** of strategy formulation. Here the debate is between planning approaches that emphasize rational design and organizational learning approaches that emphasize the emergent realization of strategy. Finally the *technical requirements* for the system can be found in modern distributed information processing **systems** that support collaborative multi-agent environments composed of both software agents and human agents.

2.1 Information Requirements

The definition of strategy and the nature of competitive advantage derived from strategy has been a source of contention in the strategic management literature for some time. Chandler's (1962) classic formulation was based on fit. Fit is both internal (the capabilities of the organization to its structure) and external (the fit of the organization's structure to the environment). This idea of fit leads to the classic SWOT model (strengths, weaknesses, opportunities, threats) for strategic analysis of the firm's external environment (OT) and the internal resource capabilities of the firm (SW). The fit between the environment of the firm and the capabilities of the firm remains to the present day in the strategy-structure-performance view of strategy exemplified by Porter (1985) and the resource-based view of the firm exemplified by

Barney (1991). The debate between the resource-based view of the firm and the competitive forces view of strategy and competitive advantage is a debate about the content and focus of strategy.

The central content themes of strategic theory provide a basis for deriving the information requirements for a strategic decision support system. If strategic management information systems are to become more than "management misinformation systems," it is important that the information they deliver is relevant to the task of strategic decision making and not a system that delivers an "*over-abundance of irrelevant information*" (Ackoff, 1967). The premise of this dissertation is that, for a SDSS to be effective, it must look to the nature of strategy for a determination of what information a strategic support system should deliver to management decision makers.

Information requirements for decision making are derived (or should be derived) from the management control system of interest. Three types of management decisions relevant to analyzing the information requirements for a decision support system (Ackoff, 1967) are:

- Decisions for which adequate models are available or can be constructed and from which optimal (or near optimal) solutions can be derived.
- Decisions for which adequate models can be constructed but from which optimal solutions cannot be extracted.
- Decisions for which adequate models cannot be constructed.

Of these three types of decisions, the third most accurately describes strategic decision making. Research is required to determine the information requirements for this type of decision making (Ackoff, 1967). Such research will of necessity apply to the specific problem at hand, but a framework for strategic information requirements can help take some of the guesswork out of the analysis and provide a generic model that is amenable to solution. In this context a review of strategic management literature and strategic theory is appropriate in that it can provide a basis for building an information requirements framework.

Much of the strategy literature revolves around two classic approaches to strategy: The analysis of competitive market forces (Porter, 1980; Porter, 1985) and the Resource-Based View **(RBV)** of the firm (Barney, 1991). More recent extensions of RBV have proposed the necessity of firms developing dynamic capabilities in the face of rapid technological change (Teece, Pisano & Shuen, 1997) and "grand unified theories" that integrate the other streams into a single theoretical construct (Phelan, 1997; Amit & Schoemaker, 1993). My purpose here will be to

briefly discuss the central theoretical streams and the kind of information requirements they imply for a SDSS.

2.1.1 Strategic industry factors

According to the market forces view, there are two primary factors to consider in competitive strategy: the firm and the industry in which it competes. Some industries are attractive while others are not, in that the competitive forces in that industry preclude gaining competitive advantage. To determine the difference, one must conduct a thorough analysis of the competition in the industry using the five forces model of competition: 1)inter-firm rivalry, 2) threat of new entrants, 3) customer bargaining power, 4) supplier bargaining power, and 5) potential substitutes (Porter, 1980). The firm, simply stated, is a series of activities undertaken to produce goods or services which a customer will purchase at a price that is greater than the cost of producing them. This process is described as a "value chain".

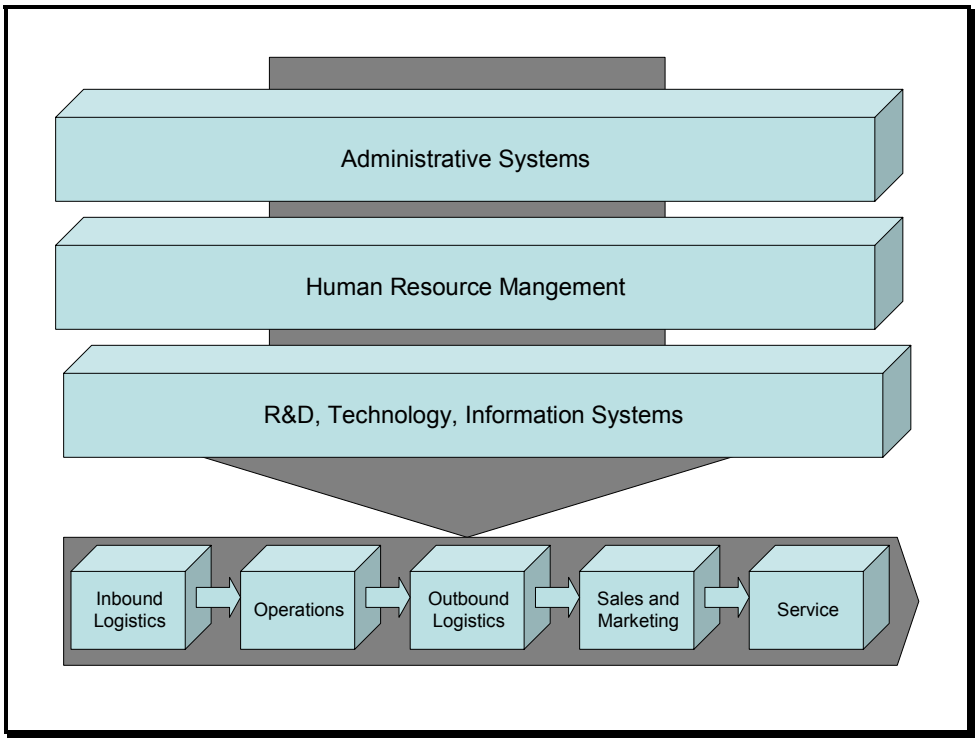


Figure 3. Porter's value chain model (Porter, 1985)

The value chain is a subset of the value system, which is the series of value chains that make up the channel that moves goods from raw material to the purchaser. Value Chains are a

central means of diagnosing and enhancing firm competitive advantage. As an analytical tool the value chain enables an understanding of competitive advantage and underlies the conceptualization of strategy (see **Figure 3**). Competitive strategy in this view is the search for the correct set of firm activities that allows for the generation of profit based on the competitor's value chain and associated industry structure. Industry structure determines which businesses will capture and retain value. Because firms are not created to realize profit on a one-time basis, the strategy must also allow for the creation of barriers to inhibit imitation. Strategies should prevent the erosion of competitive advantage by competing firms. Firms can accomplish this through unique sets of activities that distinguish one firm from another in the marketplace (Porter, 1996).

Another threat to firm profitability is the evolution of the industry in which the firm competes. As the industry ages, it evolves due to environmental changes and the activities of competing firms (Phelan, 1997). If a firm does not adjust its strategy as the industry evolves, it may be producing goods that are no longer valued by the customers. Therefore the barriers created by the firm to protect its ability to produce profit must be a "moving target" (Porter, 1985). Strategy then is the tool employed by a firm to defend itself from the economics of a free-market competitive system. As soon as an individual firm realizes an economic profit there will be another firm that attempts to gain the same advantage and thereby take away the profit, returning the market to parity. Strategy is a continual process of adjusting both the fit between the company and the market and the internal fit among the activities that make up the firm's value chain (Porter, 1996).

As part of the concept of fit, Porter (1996) emphasizes the combinatorial or complementary effect of integrated systems of activities. Systems of interlinked activities develop a value chain that is distinctive and sets a firm apart from its competition. The better the fit between the various activities of a firm the more difficult it is for competition to imitate specific actions of that firm. The whole system of activities must be imitated. This is not possible because of the required trade-offs in strategic position. Selecting integrated sets of activities involves tradeoffs. Although Porter discusses strategic fit as the integration of activities internal to the firm, implicit in his theory is the external fit to the strategic environment. For Porter there are three sources of strategic position which are not mutually exclusive and which sometimes overlap.

- Variety-based positioning – Focusing on subsets of products or services

- Needs-based positioning – Serving most or all of a particular group of customers
- Access-based positioning – segmenting customers who are accessible in different ways

Positioning involves trade-offs and fitting a unique set of activities to a particular market segment. The key is the system of activities and their fit with the strategic position in the market place. The value chain provides an important analytical tool for understanding and conceptualizing the firm's competitive environment, the process of value creation and the internal fit of firm activities to provide a unique competitive advantage. Internet technology has dramatically affected the value chain through enterprise wide systems development and integration of the supply chain and customer relationship management. End to end management of **value systems** that interlink companies is now possible (Porter, 2001).

From an information processing standpoint, this view of strategy supports the understanding of opportunities and threats **(OT)** in the market environment and the integrated activities of the value system. These strategic industry factors are key to understanding the external environmental context of the firm whick impacts firm performance and competitive advantage. Strategic decision making has traditionally taken these factors into account in formulating strategy; the information is now available in the intranet and extranet environment of linked strategic partners. This wealth of information contributes to the problem of mis-management information systems.

2.1.2 The resource-based view of the firm

The resource-based view **(RBV)** of the firm has taken a deeper look at what constitutes strategic assets in a firm that confer competitive advantage. While different adherents exist within the RBV school (Schulze, 1994), RBV is currently considered by many as one of the leading theories of strategy (Barney, 1991; Peteraf, 1993). RBV theory looks at the strategic assets of the firm that confer economic rents (Lockett & Thompson, 2001). Strategic assets are intangible assets (Hall, 1992), such as knowledge-based assets (McEvily & Chakravarthy, 2002) that provide barriers to imitation and thus confer competitive advantage (Amit & Schoemaker, 1993).

RBV theory defines the characteristics of firm resources and the role they play as a source of competitive advantage. Firm resources are defined as those physical, organizational, and human resources that can be used in value creation strategies to increase firm efficiency and

effectiveness. Competitive advantage in the resource based view is achieved when a firm is executing a value creation strategy not simultaneously being implemented by other firms. Value creation strategies are sets of actions that use selected rare strategic resources to take advantage of environmental opportunities and counter the threats of the competitive forces. Competitive advantage can only be sustained if the resources upon which it is based are inimitable, non-substitutable, and non-transferable. Sustainable competitive advantage is thereby maintained until demand shifts occur in the environment which force a change in strategy and the value of the associated strategic resources (Barney, 1991).

Both the RBV and competitive forces view of competitive advantage suffer from the same ambiguity with regards to the concepts of competitive advantage and the incorporation of firm resources in value creation strategies. Recent research has focused on unifying the two views (Priem & Butler, 2001; Barney, 2001). The RBV assumes that the context that determines the value of resources is the competitive industry environment. Value creation strategies are the link between firm resources and competitive advantage. Similarly for Porter, generic value creation strategies lead to the achievement of competitive advantage. How such strategies are conceived and implemented, how resources are cultivated and used in implementation, and how opportunities and threats in the environment are responded to can potentially help clarify the concepts of competitive advantage, the value of resources, and the selection of generic strategies.

Integration theories (Priem & Butler, 2002; Amit & Schoemaker, 1993) that bring RBV and the competitive forces paradigm together as two sides of the same coin have been proposed. Integration theories fundamentally relate Strategic Industry Factors **(SIF)** to Strategic Assets **(SA)** as the two sides of that coin. The devil, however, is in the details. Criticism (Phelan, 1997) of the approach points out that the relationship between SIFs and SAs is not clear. The nature of the fit between the two is not structurally explained. Phelan views SIFs as theoretically distinct from SAs. The overlap between SIFs and SAs is in the fact that certain resources are needed to overcome or exploit structural distortions in the marketplace. The complementary nature of the interaction between SIFs and SAs and how they contribute to value creation strategies will clarify the concept of competitive advantage and may provide a clearly articulated theory of strategy (Powell, 2001).

The integration of theory is central to Phelan's (1997) grand theory of strategy. The theory is distilled from three primary streams of strategy research: market forces, RBV, and evolutionary dynamics. The theory encompasses four key axioms.

- Firms are rent seekers.
- Strategy is a deliberate set of actions taken to create rents.
- Rent creation involves obtaining a strategic "fit" between the organization and its environment.
- The degree of fitness changes over time due to competitive and other (exogenous) influences.

The idea of strategic fit and dynamic change seems to bridge the gap between strategic assets and strategic industry factors. The source of competitive advantage is in the degree of fit of the unique activities of the firm that align internal assets with external market factors in a unique strategic position which is more than the sum of the parts. This fitness changes over time in response to the dynamics of the market place and depends on the capabilities of managers to think strategically and reposition their firms accordingly.

Is there emerging a grand theory of strategy as Phelan suggests? The dynamic capabilities extension of RBV was first proposed as a way to adjust RBV theory for situations where the rapidity of change and technical revolution in the environment invalidated the theory. How can resources alone provide a sustainable competitive advantage when a dynamic environment is constantly changing the strategic value of those resources? The search for strategic assets that confer some type of eternal competitive advantage has resulted in the invisible asset (Godfrey & Hill, 1995). It can't be comprehended, copied, imitated, substituted etc. Such strategic assets are somewhat like the Holy Grail. The Grail is not of this earth, but the campaign to find it consumes the minds and efforts of generations. The search for the disappearing asset and integration theories have lead researchers to focus on the role of management and the development of dynamic capability approaches to strategy.

2.1.3 Dynamic capabilities and the role of the manager as strategist

A recent extension of RBV is the dynamic capabilities theory of strategy (Teece et al., 1997; Makadok, 2001; Eisenhardt & Martin, 2000). Dynamic capabilities theory is introduced as a solution to the problem of dynamic change in the environment which obsolesces strategic assets. In fast changing industries and markets, today's strategic asset is tomorrow's millstone. RBV breaks down in environments that are increasingly becoming the norm for industries with large technical inputs and products. In Teece's view, firms need to develop dynamic capabilities

to continually adjust resources and develop new capabilities in response to changing market conditions.

Teece's (1997) dynamic capabilities framework focuses on managerial and organizational processes which are shaped by specific asset positions and constrained by path dependencies. Managerial and organizational processes must integrate and coordinate firm activities both internally and externally. While internal coordination has always been seen as a key role for management, external coordination of the supply chain, strategic alliances, and virtual corporations are pushing management's role to externally coordinate the actions of the firm. Information technology is a key strategic system in this newer external thrust.

The integration of IT and organizational/managerial processes demands complementary changes in the operating procedures and decision making processes of the firm. More important than integration and coordination, however, is the capability of the organization to learn and develop new routines and processes in response to changes in environmental conditions and changes in technologies. Constant surveillance of markets and technologies is needed in order to continually transform the fit between the organization and its environment.

Dynamic capabilities include management's capability for strategizing. The process of making strategy is a capability that has specific activities, processes, etc. that improves firm performance and this provides a competitive advantage (Chakravarthy, Mueller-Stewens, Lorange & Lechner Christoph, 2003). The ultimate competitive advantage comes from the management team and its ability to continually formulate the strategic problem and find solutions to it. The importance of managerial cognition in this process gives rise to cognitive approaches to strategic management which will be important when considering the **process** of strategy making.

2.1.4 Strategic theory and information requirements

Strategic theory is embedded in the process of strategy making in the SWOT analysis. SWOT analysis is frequently used as a basis for strategic decision making. It is the primary scan of the internal and external environment and assessment of the fit between SW / SA and OT / SIF. Traditional strategy making defines coordinated actions that fit the two together. Strategic theory suggests that the information requirements for a strategic control system are of two generic types: strategic industry factors and strategic assets of the firm.

Information on strategic industry factors available through current information technology is overwhelming. Two primary sources of information are the internet and corporate

extranets. The internet offers a broad selection of information on competition, new technologies that affect an industry, new entrants to the market place, product developments, and substitution threats. Intelligent scanning of the environment is a constant task that requires an intelligent system that can quickly select information given a specific model of the problem at hand. Corporate extranets provide a wealth of information on strategic partnerships and capabilities. Porter's value chain provides an excellent analytical tool for determining the scope of information requirements in this sphere.

Strategic asset information is currently being captured in the corporate intranet and includes traditional information systems capabilities (data mining, decision support systems, knowledge management systems, executive information systems, and expert systems.) The literature on the resource based view makes clear the importance of knowledge as a strategic asset. A strategic information system requires information about firm capabilities in the selection of assets for value creation strategies. Such information is increasingly available through the extensive development of information technology in firms today.

How fit is ascertained is a question that can be researched and developed as a practical application of theory. But is it just a question of fit? The difficulty is in the complexity of the integration of systems internal and external to the firm, the precise nature of the fit and the necessity to continually adjust one's strategy to changing conditions. Given the bounded-rationality of managers, this would seem to be a humanly impossible situation. Managers are not given to long deliberations on optimal strategic solutions and endless strategic planning exercises especially in a fast changing, dynamic, and uncertain environment. They have better things to do with their time. (Thus the popularity of "Intuitive" approaches to strategic thinking and the general decline of the strategic planning departments of firms.)

If the view of strategic fit is correct, a simple intuitive approach is suboptimal. The complexities of integrated activities in the value chain, of fitting one's strategic position in the environment, and of keeping on top of rapidly changing needs and technologies in the market place makes it difficult for even the brightest manager to make the right call. The team approach puts more eyes on the situation, but more blind men do not entirely describe this particular elephant. Quantitative modeling and simulation approaches have been very effective at the tactical level in solving complex problems once the problems have been formulated. This is a time consuming process, however, and application to the strategic level of decision making has a number of drawbacks, especially since the techniques assume that a problem has been formulated

that is amenable to computational solutions. The key question here is what information is relevant? A frame of reference is needed. Ackoff (1967) suggests that the implicit decision models of managers need to be made explicit and provide the basis of determination.

The nature of strategic fit is a larger problem space with more complexities. Tactical solutions are by nature limited in context. Yet tactics taken as a sum do not a strategy make. The same set of tactics can be used to implement different strategies just as the same paint palette can be used to make different paintings. Timing, order, and combinatorial effects make strategy both art and science. The strategic context makes sense of and integrates specific tactical approaches into a unique approach that is more than the sum of the parts. Strategic decisions are systemic decisions of Ackoff's (1981) third type. The problem situation is unstructured. The uncertainty associated with the environment and the complex interaction between environment and the strategic actions of the firm suggest a different dichotomy is at play in the development of strategy which is a combination of both rational design and the emergence of strategic patterns resulting from the implementation of strategies.

The abundance of information both external and internal to the firm that is currently available to management increases the probability of "management misinformation" (Ackoff, 1967). What is lacking in today's information rich environment is a context for selection of relevant information: the management team's implicit mental model made explicit. This modeling needs to be part of a knowledge management system that intelligently and proactively filters information on strategic industry factors and strategic assets. Research in strategic theory needs to guide practice in this area.

2.2 Processing Requirements

The processing requirements for an SDSS can also be found in the strategic management literature on the strategy process. The **process** of strategy making has historically been a debate between strategy as rational design and strategy as emergent through incremental realization. The merits of the design school versus the incremental approach have been the subject of much debate in the literature (Mintzberg, 1990; Ansoff, 1991; Mintzberg, 1991). Mintzberg (1990) critiques the "design school" approach to strategy epitomized by the Harvard Business School and the work of Michael Porter while Ansoff defends the basic premises of a rational approach to strategic planning. Two different approaches to the process of strategy making are identified (Mintzberg, Lampel, Quinn & Sumantra, 2003):

- *Strategy is a plan*. Strategy is a deliberate and intentional plan that may also involve gaming the competition with maneuvers and ploys. The formulation of strategy as a plan is a top down rational process of formulation. This is the *a priori* approach to strategy formation: strategic action is a logical consequence of a rational design.
- *Strategy is an emergent pattern*. Strategy is a consistent pattern of behavior on the part of organizations. This view of strategy focuses on realized strategy, the strategy that actually emerges from both intended and unintended actions of an organization in pursuing its goals. The process is a bottom-up organizational-learning approach to strategy. This is the *a posteriori* approach to strategy in which strategy is a logical consequence of experience and learning.

2.2.1 Design approach: strategic planning

This approach is a top-down approach of intelligent design. Strategy is explicitly conceptualized by the top management team. It is prescriptive and provides the central theme for tactical actions that implement conceptualized strategies. Perhaps the model that most epitomizes the design school is the familiar SWOT model. This model is embedded in strategic planning processes that see strategy as a calibrated fit between the external environment and internal structure or capabilities. The external environment and internal capabilities are appraised and strategies are designed to maximize the fit between firm environment and capabilities. The process of strategy making is analytical and follows Simon's problem solving model (Fernandes & Simon, 1999) of intelligence (SWOT analysis), design (alternative strategies are created), and choice (strategies are evaluated and an optimal one is selected).

Mintzberg (1990) characterizes the design school approach to strategizing:

- Strategizing is a controlled conscious thought process
- The CEO is the chief strategist in control of the process
- A strategy is a set of simple conceptual models held by the CEO
- Strategies are unique sets of activities that distinguish a firm from its competition
- The process defines fully formulated strategies that allow clear choices
- Explicit strategies are of necessity simple and aid the communication process
- Strategy formulation precedes implementation

Mintzberg (1990) argues that this approach is flawed because experience precedes thought. A firm can only analyze strengths and weaknesses in light of the success or failure of its

concrete actions in the environment. Structure (organizational capabilities) does not follow strategy but is concurrent with strategy. Explicit strategy promotes inflexibility and blinds managers by creating cognitive bias and investment in preferred ways of doing things that limits strategic flexibility. The design school separates thinking from doing by insisting that thinking take place first and doing follow thinking. This is implemented in a machine bureaucracy which systematizes decision making into a top-down hierarchical process which ignores the experience and learning of individuals at the operational end of the business. Organizational learning is seen as mere opportunism. The inflexibility of the rational approach breaks down under the uncertainty and turbulence found in most situations.

Ansoff (1991) counters Mintzberg's (1990) characterization as an oversimplification of design school premises and that the school has evolved toward "holistic strategic management." Holistic strategic management is a systems approach to strategic management that involves structured methods for analytic strategy formulation by more than just the CEO. It identifies specific procedures that counter problems of cognitive bias, and it examines the overall process of organizational learning and adaptation to turbulent environments with a rational approach to learning.

2.2.2 Incremental approach: emergent strategy

Mintzberg (1994) suggests the design school relies on an analytical approach to strategic planning more appropriately relegated to the implementation stage once strategies are formulated. He posits that strategy formation is synthetic, and therefore an intuitive and creative process, and that analytical thinking and structured approaches to planning can actually get in the way of strategic thinking and be counter-productive in developing strategy that is then programmed by strategic planners.

Ansoff (1991) summarizes Mintzberg's learning approach:

- Strategy is emergent and based on experiential learning
- Strategy is the logical and observable pattern perceived *ex post*
- Explicit strategies lead to cognitive bias (blinders)
- The unpredictability of turbulent environments invalidates *ex ante* strategic choice
- Knowledge of predictable environments comes from experience
- Future forecasts are unreliable

- Managers shouldn't make forecasts
- Strengths and weaknesses are determined by experience
- Organization-wide orchestration of explicit strategies is impossible
- The design approach is only useful for startups and periods of operating stability

Ansoff (1991) critiques this approach by pointing out that this view of strategy is at odds with the reality of modern strategic management. In turbulent environments, businesses that depend on learning first will lose out to those who have foresight and planning which enables them to capture market share. Mintzberg's (2003) organizational learning is reactive not pro-active. Firms that use a trial and error approach to strategic management will inevitably experience more failure than success and the cost of failure can be prohibitive. The uncertainty of the environment means that more explicit *a priori* strategy formulation is needed, not less. In practice firms formulate strategy in advance of implementation. The trial and error process of strategy formation is disconnected from cognitive reflection that is used to inform explicit *a priori* strategy formulation. Incrementalism uses an existential learning model that excludes the advantages of rational thought. A rational model of learning uses cognition to optimize the selection of alternatives which is less time consuming, anticipates the future, and reduces costly errors.

The Mintzberg (1990; 1991) – Ansoff (1991) debate exposes the gap in the strategic process literature. On one side is design with its rational learning model and on the other emergence with its existential learning model. Strategic control is a sophisticated rational learning model that utilizes a "cognition – trial – cognition – trial" looped learning model. Mintzberg concedes this point but leaves open the question of whether rational analytical approaches to the synthetic creative process of strategy conceptualization are possible. Mintzberg's critique has led to a further examination of design school and related planning school premises. The interplay between learning and planning is central to managing organizations in today's turbulent and uncertain environments. Strategic management is a process of continually adjusting the fit between the firm's capabilities and its changing environment.

2.2.3 Closing the gap through planned organizational learning

Can the gap be closed? The systems approach provides a holistic solution that is inclusive of both rational and existential approaches to learning. A systems approach posits

planned organizational learning (Georgantzas & Acar, 1995) that integrates planning and learning in a holistic approach to strategic management. This approach embraces the gap as an opportunity for organizational innovation and creativity through incorporation of new management technologies. Among these new technologies are technologies of participation (Spencer, 1989; Wilson, Harnish & Wright, 2003), cognitive mapping (Eden & Ackermann, 1998), and scenario-driven planning (Georgantzas & Acar, 1995). These approaches are systems approaches that integrate rational and existential learning in a process that intentionally structures the organizational learning process of cognition – trial – cognition – trial.

2.2.3.1 Technology of participation

Managing the process of change in corporations is the focus of technologies of participation. This approach utilizes both rational modeling and existential learning in a structured approach to organizational learning and change based on the work of the Institute of Cultural Affairs (ICA), a global research, training and demonstration organization. The approach uses a highly structured process model of participation and interaction (a technology of participation) in formulating strategy (Spencer, 1989). Facilitation methods structure and enable creative cognitive processes that build consensus through the creation of inclusive cognitive frameworks which blend multiple perspectives into unified systems models. This stands in marked contrast to consensus processes which force agreement through voting (nominal grouping technique or expert opinion such as the Delphi technique (Druckenmiller, 1991). These methods have been extensively tested and refined in local development projects, community meetings, and corporate strategy sessions in thousands of decision situations from the mid 1970's to the present.

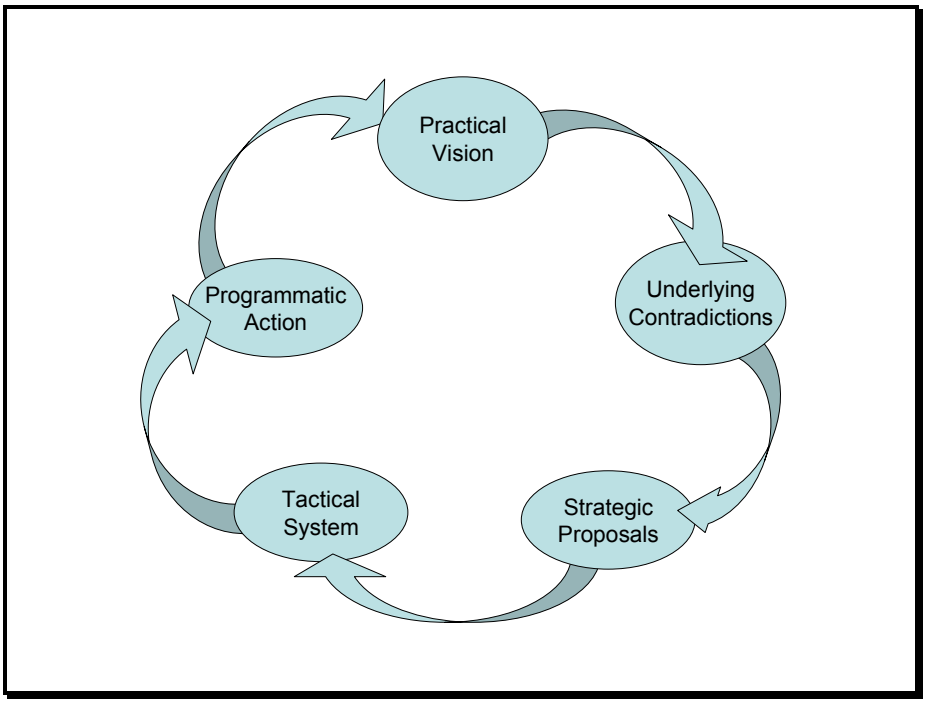


Figure 4. The ICA's strategic planning process

Figure 4 illustrates the strategic planning process of the ICA (Spencer, 1989). The process is a planning and implementation loop that implements the cognitive-trial-cognitive-trial process of organizational learning. A key innovation of the process is found in the practical vision and underlying contradictions components. A practical vision provides a vivid idealized picture of the future in some detail. This is in contrast to vision and mission statements found in traditional strategic planning processes. It is a composite integrated framework of the envisioned future that represents the aspirations of the participant decision makers. The vision is specific and within a timeframe that represents the strategic planning horizon. In contrast to the practical vision are the underlying contradictions. These represent the current reality of **structural** impediments to the practical vision. Contradictions are the underlying network of root causes behind observed symptomatic problems. They must be taken into consideration as a whole for strategies to be effective in sustaining change from the current reality to the practical vision.

Looking at Figure 5, the gap between the practical vision (which is an idealized design of the future (Acar, 1983)) and the current reality results in a creative tension that forms the basic crucible out of which strategic proposals are forged (Wilson et al., 2003). Strategies are multiple,

simultaneous courses of actions that resolve the contradictions and move the current situation toward the practical vision of the future. A tactical system is then designed which implements the proposed strategies, and tactics are grouped into programmatic actions around which resources are committed. Implementation of programmed actions is evaluated using a structured process of reflection and the cycle repeats itself with a new round of vision, contradictions, etc. This cycle of organizational learning uses explicit rational models as a tool for organizing experiential learning in a continuous improvement process that is both participatory and structured.

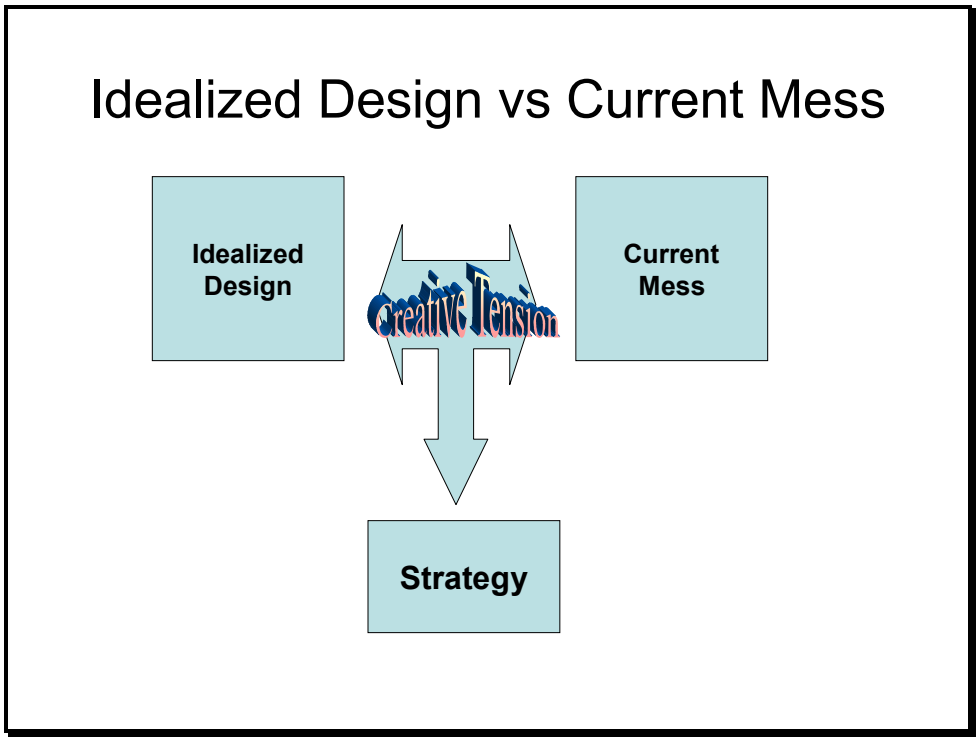


Figure 5. Gap analysis and the formation of strategy

2.2.3.2 Cognitive mapping

Eden's approach to strategy making also uses a continuous cycle of learning through a review of both emergent and planned strategies. The process is an integration of a variety of schools of strategy and is described as a journey of strategy making (Eden & Ackermann, 1998).

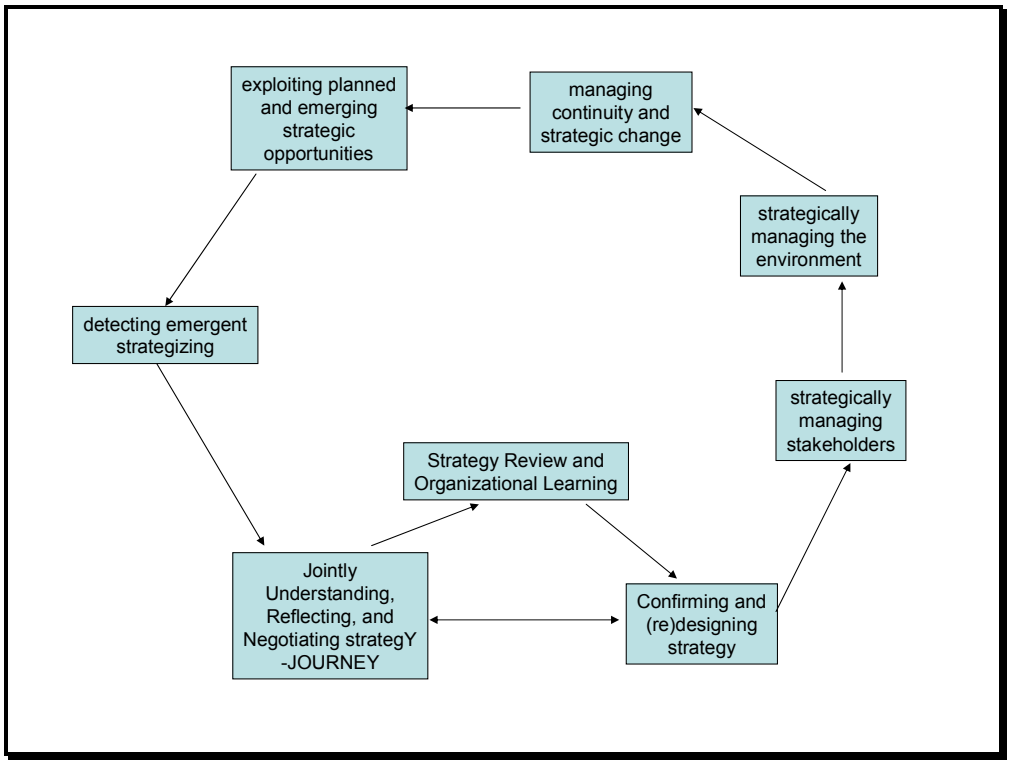


Figure 6. Eden's strategy making process

The process is illustrated in Figure 6. The cycle represented in the figure represents a journey of strategy making. The process is participatory and builds a shared perspective and consensus through an integration of individual perceptions. A deeper understanding is gained of strategic assumptions and the dynamic interplay of individual belief structures. This deep knowledge promotes organizational learning. The inner loop or cycle is a rational process of organizational learning through review that redesigns current strategy based on experience. The outer loop is the process of emergent strategy through experiential learning that feeds back into the review process. The strategy making process thus deliberately incorporates both rational and existential aspects of organizational learning (Eden & Ackermann, 1998).

Eden's approach to strategy making was developed initially in the late 1970's and as a qualitative modeling approach to understanding complex and messy organizational problems with political and cultural dimensions. This was in contrast to the quantitative approaches of management science **(MS)** and OR that were being used to solve well-structured problems through mathematical modeling. Eden developed an explicit cognitive approach based on a

cognitive mapping technique in a group facilitation framework. A cognitive map is a graphic representation of individual perceptions and beliefs about a situation.

Graphic representations are well known in the management and social science literature as both systems analysis tools and knowledge representation techniques (Sowa & Dietz, 1999). A number of specific approaches have been developed, including adaptations of mathematical graph theory (Harary, Norman & Cartwright, 1965), cognitive psychology and *personal construct theory*, (Kelly, 1955) *influence diagramming and causal mapping,* (Maruyama, 1963) and *systems dynamics* (Forrester, 1961).

Huff (1990) identifies specific classes of cognitive maps and Eden's (1998; 2004) work in this domain is considerable. Initially focused on problem formulation, cognitive mapping began to be applied to strategic problem formulation. A general theory of strategic problem solving was developed that incorporated cognitive mapping with a facilitation process that was computer supported. This Group Decision Support system is a designed methodology called Strategic Options Development and Analysis **(SODA)** (Eden & Ackermann, 1998).

Cognitive approaches have focused on the challenge of finding analytical approaches to the synthetic and creative process of strategy conceptualization. Intuition can be informed and creative processes can be structured. This approach integrates the emergent strategies of Mintzberg (2003) with the deliberately rational learning approach advocated by Ansoff (1965; 1991). Cognition is important in confirming and redesigning strategy, for managing complexity and organizational change, for strategic decision making and problem solving, and for enabling cognitive change and flexibility (Eden & Ackermann, 1998).

2.2.3.3 Scenario driven planning

Eden's work exposed a gap between problem framing which depends on qualitative analysis and problem solving which depends on quantitative analysis. Problem framing seeks to understand a focal strategic situation and through cognitive processes develops strategic models. Structured facilitation methods which support cognitive processing are well developed and supported by information technology through group decision support systems. Quantitative analytical approaches use mathematical models to arrive at solution sets through mathematical modeling. This includes 'hard' approaches such as systems dynamics (Forrester, 1961) and classical decision theory and the field of OR/MS. Quantitative analytics are also well supported by information technology with decision support systems.

Scenario-driven planning closes the gap exposed by Eden by blending qualitative and quantitative analytics into a unified methodology (Georgantzas & Acar, 1995). Scenarios have long been used to support strategic thinking and are a key tool in dealing with "wicked problems" (Mason & Mitroff, 1981). Strategy and policy situations are real-world problems that exhibit complexity in that they are composed of many interrelated problems and issues. To be effective, strategies must holistically address the complexity of the situation rather than propose solutions to single problems. Formulating and understanding the situation and its complex dynamics, therefore, is key to finding holistic solutions. A systems approach to problem formulation stresses that single problems cannot be isolated from surrounding messy realities. The messiness of reality requires a shift from problem formulation to situation formulation (Ackoff, 1981). The analysis of **change scenarios** (Schoemaker, 2002) allows the design of strategies to take place in the messiness of the situation.

CSM is a powerful tool for scenario driven planning developed by Acar (1983). It endows causal mapping with rich computational properties. By including in the method indications not only of the signs of the presumed causal influences, but also of their intensities and the possible time lags, Acar developed a technique for simulating manually (on the causal map itself) the propagation of change through a causal network. The rich computational semantics of Acar's causal mapping approach support automated modeling and simulation in ways that other varieties of cognitive mapping and causal mapping do not. CSM is a powerful analytical tool that allows the "forward analysis" of a situation in the sense of computing its implications as to the kinds of potential change scenarios that might occur (Acar, 1983; Georgantzas & Acar, 1995). Acar's causal mapping approach was originally designed as a manual system because of this. While these assumptions held true in the early 80's, improvements in information technology and the proliferation of DSS, GDSS and GUI object-oriented techniques of development forced a re-examination of those design constraints. A complete description of this approach to strategy making is the subject of Chapter 3.

2.2.4 Strategic theory and process requirements

The process of strategy making is an interactive, dialectical process that surfaces the emergent strategic assumptions present in an organization and allows them to become an explicit part of a rational strategic plan. An explicit rational strategic plan allows for an idealized design of the future, but takes into consideration the messiness of current reality. The inputs to the

process are information about the current environment and the state of the organization's capabilities as revealed in its interactions with the environment. Cognitive perspectives are made rationally explicit as an aid to a systematic reflection on strategic position and a guide for strategic control. The process of strategy making is continual and ongoing, takes place in real time, and is based on existential and rational learning. This indicates a systems approach to analysis and design of strategy that is integrative, holistic, and inter-subjective.

2.3 Technical Requirements

The analysis of the information and processing requirements paints a picture of a multi-agent systems environment. Multi-agent systems are distributed, complex interactive socio-technical systems that accomplish shared goals and purposes (Chaib-draa, 2002). The technical environment supporting such a strategic decision support system needs to utilize technology that is platform independent, is distributable through common internet technology, and is integrative of both human and software systems of decision making. This type of system is a multi-agent inquiring system **(MAIS).** A MAIS is a knowledge blackboard system that autonomously and intelligently facilitates the inquiry of a community of inquirers who share a common ontology[1] (Ugwu, Anumba & Thorpe, 2001). The ontology of the community is represented in a form that intelligent agents can understand and is used to coordinate the activities of the agents in solving problems of interest to the community.

2.3.1 The design of multi-agent inquiring systems

MAIS is a **theory of the blackboard** which is a theory of knowledge representation that integrates the design of inquiring systems (Churchman, 1971), knowledge discovery by artificial intelligence systems (Simon, 1998), personal knowledge systems (Boulding, 1969), and automated analogy solvers (Hofstadter, 1996). In particular Churchman's work has had a key role and influence in the conceptual foundation of MIS research.

[1] In philosophy Ontology is the study of being, things that exist. In knowledge engineering circles and the artificial intelligence community an ontology is a shared explicit description of the things that exist in the world of discourse. Agents need a shared ontology to sensibly communicate with each other.

In the ***Design of Inquiring Systems*** Churchman (1971) identifies several primary types of inquiring systems. He proposes an intelligent assistant that can take over the research task and raises the question as to where the boundary is between human and machine intelligence, what roles and tasks are most efficiently performed by intelligent assistants, and which must be left to the human inquirer.

Herbert Simon (1998), on the other hand, proposes that all inquiry tasks including discovery science can be taken over by intelligent assistants and devoted much of his career to developing computer applications to prove his theory. In his paper on scientific discovery, he discusses the role of knowledge representation and analogies in developing scientific theory. Simon pushes the envelope as to how much may be accomplished by intelligent assistants. From Simon comes an understanding of the hierarchical levels of knowledge representation and a correspondence with the work of Churchman.

Douglas Hofstadter's (1996) work on fluid concepts and analogies applies distributed artificial intelligence approaches to the problem of modeling human analogical reasoning and problem solving. Analogical reasoning is identified by Simon as a key to the knowledge discovery process. Developing artificial intelligence systems that reason analogically provides for robust matching capabilities where the fit between the conceptual constructs of humans and matching information or constructs cannot be determined by literal interpretations of semantic content.

Kenneth E. Boulding (1969) in his seminal work: ***The Image: Knowledge in Life and Society*** provides another view of the blackboard. Boulding, an economist from the University of Michigan, proposed a new interdisciplinary science based on his theory of the Image. He called the science Eiconics. This new discipline (1956) would be based on the new interdisciplinary science of cybernetics, information theory, and analytical psychiatry. At the time it was part of the new inter-disciplinary movement in academe.

Boulding (1969) conceptualizes social frameworks and hierarchies of knowledge through his theory of the image. The image is a knowledge structure that exists at both the individual and public level. Subcultures, or sub-components of the public image such as scientific disciplines, have images as well that are transcripts of the interplay between the image and value structures of individuals in that subculture. Each individual's reading of the transcript is processed through his own value structure and is incorporated into his personal image. Messages to the public image are processed through its value structure (a sub-component that is itself a collection of

individuals) and potentially become part of the structure of knowledge at that level. Messages are frequently rejected by value structures or can be modified to conform to the value structure. If a message is sufficiently powerful, it can be forced through the value structure to occasion change in the image including the value structure itself. The process of learning is essentially a process of image reorganization.

Churchman places emphasis on the integration of the components that make up an inquiring system. No one part can be designed without consideration of the whole. Strategic management is the designer, decision-maker, and user of the system. Simon says the system is a complex hierarchy that is entirely designable as an automated artificial intelligence system including knowledge discovery. Hofstader provides a model of how distributed AI can accomplish this task. Boulding says that the system is a sociological knowledge structure that can be understood as an image. This is the transcript of knowledge that managers contribute to and draw inspiration from in the inquiry process. The blackboard is embedded in a sociological structure of inquiry.

A strategic decision support system is a practical application of MAIS. It implements a system of inquiry that models human cognition and knowledge of strategic situations. This is of necessity a collaborative dialectical process that develops the inter-subjective agreement identified by Churchman as a Singerian system of inquiry. Development of Singerian systems is now possible with new developments in AI research. Research into DAI and in particular MAS design is the basis for the technical development of a strategic decision support system.

2.3.2 Distributed artificial intelligence approaches

Distributed artificial intelligence has historically been divided between distributed problem solving **(DPS)** and MAS. The later are societies of agents that collaborate to solve complex real-world problems. Multi-agent systems have been around for some time as a distributed artificial intelligence research topic. There is little consensus as to what constitutes an agent, but the following characterization is useful (Jennings, 2000):

> An agent is an encapsulated computer system that is situated in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectives.
>
> There are a number of points about this definition that require further explanation. Agents are:

> - clearly identifiable problem solving entities with well-defined boundaries and interfaces;
>
> - situated (embedded) in a particular environment — they receive inputs related to the state of their environment through sensors and they act on the environment through effectors;
>
> - designed to fulfill a specific purpose — they have particular objectives (goals) to achieve;
>
> - autonomous — they have control both over their internal state and over their own behavior;
>
> - capable of exhibiting flexible problem solving behavior in pursuit of their design objectives — they need to be both reactive (able to respond in a timely fashion to changes that occur in their environment) and reactive [sic. He means proactive] (able to act in anticipation of future goals) (p. 280).

In a nutshell, while DPS focuses on how a particular problem can be distributed among a number of modules (nodes), research in MAS is concerned with the behavior of a collection of autonomous agents which work together to solve problems beyond the individual capabilities or knowledge of each problem solver. The characteristics of MAS are:

- each agent has incomplete information or capabilities for solving the problem, thus each agent has a limited viewpoint;
- there is no global system control;
- data are decentralized; and
- computation is asynchronous (Jennings, Sycara & Wooldridge, 1998).

2.3.3 Design methodology

Design methodology is a subject of research in the MAS research community. It is one of the critical factors that will allow it to move to the third stage in software engineering paradigms: entering the mainstream. Most approaches have been adaptations of object-oriented design (Wooldridge, Jennings & Kinny, 2000). Object-oriented design is not appropriate to agent systems because it does not translate well to the abstraction. The object-oriented design paradigm and other well-know design paradigms may be used in implementation design and are appropriate to consider at that level.

This dissertation gives credence to Jennings's (2000) perspective and builds on his work. He argues that agent-oriented software is extremely well suited to solving complex real-world system problems. He also quotes Grady Booch (originator of the object-oriented design paradigm) as identifying three tools for tackling complexity in software:

> *Decomposition:* The most basic technique for tackling large problems is to divide them into smaller, more manageable chunks each of which can then be dealt with in relative isolation. This helps tackle complexity because it limits the designer's scope; at any given instant only a portion of the problem needs to be considered.
>
> *Abstraction:* The process of defining a simplified model of the system that emphasizes some of the details or properties, while suppressing others. Again, this technique works because it limits the designer's scope of interest at a given time. Attention can be focused on the salient aspects of the problem, at the expense of the less relevant details.
>
> *Organization:* The process of identifying and managing the interrelationships between the various problem solving components. The ability to specify and enact organizational relationships helps designers tackle complexity in two ways. Firstly, by enabling a number of basic components to be grouped together and treated as a higher-level unit of analysis (e.g., the individual components of a subsystem can be treated as a single coherent unit by the parent system). Secondly, by providing a means of describing the high-level relationships between various units (e.g., a number of components may work together (cooperate) to provide a particular functionality). (p. 282)

The agent-oriented approach has merit in three fundamental areas: agent-oriented decomposition, abstraction and organization. The merits of agent-oriented decomposition are the autonomy and flexibility of control of the problem-solving entities. Because distinct agents have their own thread of control, they are well suited to complex distributed environments that have multiple loci of control. Autonomous agents can flexibly interact at runtime to react to the actual state of affairs instead of waiting for a top level controller's perception of that state. This is particularly important in distributed processing situations where no *a priori* solution for a problem exists due to its complexity and the unpredictability of the interactions of the system.

Second the suitability of agent-oriented abstractions derives from their ability to "minimize the semantic distance between the units of analysis that are intuitively used to conceptualize the problem and the constructs present in the solution paradigm" (Jennings, 2000, p. 286). For example, let's say we want to create an automated system to support a strategic decision making process. In our analysis of the process there exist many different decision makers who interact with each other in a team environment to formulate a strategic situation.

Each actor has a set of assumptions about the situation that can be formulated as a causal map of key factors and their relationships. It is convenient therefore to use these actors and their conceptual models as units of analysis. Using an agent-oriented approach to development, an agent system can be defined with agents representing each actor. The agent would manage the interface between the manager and the system and the manager's conceptualization of the situation represented in a causal map. Likewise the conceptual model of the actor can also be formulated as a set of agents that perform various tasks and interact with each other to simulate the model, etc. There is a clear correspondence between systems, subsystems, and agent organizations. Different types of agents can model different levels of abstraction. The interaction between the subsystems and their components can be easily modeled by agent systems that coordinate their actions to solve common problems or achieve common objectives.

Third the agent-oriented approach allows for flexible management of dynamic organizational structures. Agent systems have the ability to flexibly form, maintain or disband organizations. They have representational power because of their ability to be viewed as single entities at one conceptual level and as an organization of components at another. There is stability in the intermediate forms of agent systems (i.e. in the agents themselves as autonomous units) which allows them to be developed independently, so that functionality can be added incrementally in the development of a system.

2.3.4 Agent-based modeling and simulation

Like mathematical modeling, simulation forces a precise description of variables of interest and their relationships. Implicit assumptions must be made explicit, concepts must be operationalized, and formal relationships between variables must be detailed so that the explanatory mechanisms between them are explicitly formulated. Simulations can thus be used to do detailed experimental testing of theoretical formulations. Simulations can be broadly typed as to **purpose** and **approach** (Phelan, 1997).

The **purpose** of simulations is either *exploratory* or *confirmatory*. *Exploratory* simulations seek to explore relationships between system components and the observed overall behavior of the system. The system should reflect real world systems of interest and allow the user to test theoretical assumptions and hypotheses, determine the sensitivity of variables to inputs, and assess the implications of outputs on variables of interest. Face validity is more

important for exploratory simulations in that the model needs to be conceptually representative of the real world.

Confirmatory simulations, on the other hand, attempt to optimize an output variable of interest. Common to these types of simulation systems are industrial simulations or systems dynamics simulations that seek optimal solutions to identified problems which are analytically intractable or have no closed form solution.

The **approach** of simulation systems is either *macro* or *micro* level analysis. *Macro-level* simulations model situations from the top down. They specify the interactions between system components and use difference or differential equations to model systems at high levels of aggregation. Systems dynamics uses this approach. *Micro-level* simulations model individual behavior and resulting macro level behavior is emergent from the complex interactions of individual actions. A variety of agent-based modeling and simulation systems use this approach to simulation. Micro-level simulations are generally described as bottom up approaches to simulation while Macro-level simulations are top down formulations.

One of the more recent applications of multi-agent systems is in the area of agent based modeling and simulation. Originally applied to biological and ecological systems for modeling complex adaptive systems, the technique is being applied to the social sciences as well. Agent-based modeling is being embraced by a growing number of scientific researchers in a wide variety of natural and social science disciplines. In agent-based modeling, researchers create artificial worlds that model real world environments. Agents are used to populate these worlds and, based on simple rules, simulate the behavior of their real world counterparts. The researchers then use these simulated worlds to test theoretical and empirical constructs (Panepento, 2000).

An alternative approach to modeling sociological phenomenon is to use adaptive agents who interact with each other instead of static variables. Agent-based models demonstrate how simple rules of interaction between simple components result in complex, emergent patterns of composite entities. Emergent patterns are dynamic and representative of the dynamism of social environments. ABMs provide unexpected insights into holistic patterns based on the dynamic interactions of simple components.

Two basic research questions are pursued through ABM simulation: empirical evaluation of system dynamics and development of new “things that ‘ought’ to work” (Daniels, 1999). Both of these questions are relevant to scenario planning and strategic decision support. The interaction

of autonomous entities in a common environment is typically a mutually recursive process that is analytically intractable. The conceptual causal maps that are the subject of this dissertation model such environments. Simulation of complex systems allows evaluation and observation of global behaviors and system dynamics that cannot be analytically predicted *ex ante.* Similarly, simulation assists the design of new complex systems composed of multiple interacting agents. Strategy is ultimately concerned with designing such complex systems (Teece et al., 1997; Phelan, 1997).

2.3.5 Agent-oriented analysis and design

Agent-oriented design should be seen as a next step in the evolution of software engineering from machine specific design to abstractions that move ever closer to a natural description of the problem and its solution. The trend is toward more encapsulation of functionality in software building blocks. Objects encapsulate both properties and methods, agents encapsulate objects and threads of control. Agents also promote the concept of reuse in that whole subsystems or architectures can be designed to be reusable. Agents also enable legacy systems (non-agent systems) to be encapsulated in an "agent wrapper" and as such may be implemented as an integration technology (Jennings, 2000; Sycara, Decker, Pannu, Williamson & Zeng, 1996). "The distinction between objects and agents can be summarised [sic.] in the following slogan: *Objects do it for free; agents do it for money*" (Jennings et al., 1998, p. 15).

Agent systems do have their problems, however (Jennings, 2000). In particular they suffer from two major drawbacks:

> The patterns and the outcomes of the interactions are inherently unpredictable;
>
> Predicting the behavior of the overall system based on its constituent components is extremely difficult (sometimes impossible) because of the strong possibility of emergent behaviour (p. 289)

Using interaction protocols can control these drawbacks, but such protocols can limit the benefits of agent systems. A key area of research for MAS is in gaining a better understanding "of the impact of sociality and organizational context on an individual's behavior and of the symbiotic link between the behaviour of the individual agents and that of the overall system." (p. 290)

In order to tackle this problem, Jennings (2000) proposes an expansion of Newell's knowledge level analysis[2] in elucidating individual (asocial) agent principles and structures to a new social level is proposed.

> …the social level aids the science of agent-based computing by providing a comprehensive model for specifying and understanding behaviour in agent-based systems. This contrasts with the majority of the extant work in this area that typically concentrates on a small fragment of the overall picture… A social level perspective also aids the engineering aspects of agent-based systems. By identifying the key constituent components and their interrelationships, the social level provides a sound basis for developing tools that can support the development of agent-based systems. Moreover, social level models provide a basis for agent-oriented analysis and design. (p. 293)

To this end a new method of agent-oriented analysis and design, the "Gaia Method" (Wooldridge et al., 2000), is proposed. The Gaia method is a two phase iterative process: analysis and design. The analysis stage is summarized as:

> 1. Identify the roles in the system. Roles in a system will typically correspond to: individuals, either within an organisation or acting independently; departments within an organisation; or organisations themselves.
>
> *Output*: A prototypical roles model-a list of the key roles that occur in the system, each with an informal, unelaborated description.
>
> 2. For each role, identify and document the associated *protocols*. Protocols are the patterns of interaction that occur in the system between the various roles. For example, a protocol may correspond to an agent in the role of BUYER submitting a bid to another agent in the role of SELLER.
>
> *Output*: An interaction model, which captures the recurring patterns of inter-role interaction.
>
> 3. Using the protocol model as a basis, elaborate the roles model.
>
> *Output*: A fully elaborated roles model, which documents the key roles occurring in the system, their permissions and responsibilities, together with the protocols and activities in which they participate.
>
> 4. Iterate stages (1) - (3). (p. 295)

[2] see (Newell, 1982) The same Newell was credited by Churchman as having coined the term "intelligent technician" which plays so prominent a role in ***The Design of Inquiring systems.***

Once the analysis has been completed the design process transforms the abstract models of analysis into a design abstraction that uses standard object-oriented techniques of design for agent implementation. The design stage is summarized as (Wooldridge et al., 2000):

1. Create an *agent model*:
    - aggregate roles into agent types, and refine to form an agent type hierarchy;
    - document the instances of each agent type using instance annotations.
2. Develop a *services model*, by examining activities, protocols, and safety and live-ness properties of roles.
3. Develop an *acquaintance model* from the interaction model and agent model. (p. 298)

The design stage leads to a prototyping approach that allows for more detailed design and implementation. Various software platforms for prototype development are available and support the development of an agent-based approach to strategic decision support.

CHAPTER 3 AN AGENT ARCHITECTURE FOR COMPREHENSIVE SITUATIONAL MAPPING (CSM)

CSM is both a collaborative process and an analytical framework. As a process it can be viewed as a collaborative dialectical conversation that through negotiation develops a common conceptual framework (causal map) of the problem situation implemented as a blackboard structure. As such, it extends the process of "assumptional analysis" described by Churchman (1968) and Mason & Mitroff (1981). In addition, because of its forward analysis capability, the full CSM process allows for goal mapping and development of scenarios that aid strategic planning and formulation.

3.1 CSM Dialectial Inquiry

CSM is a dialectical inquiry and scenario planning method proposed by Acar (1983) that allows *backward analysis* of key assumptions and *forward analysis* of scenarios to aid strategic management in strategic thinking and foresight development. It represents an integrated approach that combines the rigor of OR modeling approaches with group process technologies which support situational analysis and problem formulation (Heintz & Acar, 1994).

The method was developed as a bridge between the quantitative, solutions oriented approach of OR and the qualitative, problem formulation approach of process consulting (Georgantzas & Acar, 1995). Critical to any problem-solving situation is the formulation of the problem. The strengths of OR approaches lie in quantitative solutions to formulated problems. The difficulty, hence, is no longer in solving problems using quantitative methods, but in formulating the problem to which various OR methods are applied (See Figure 7).

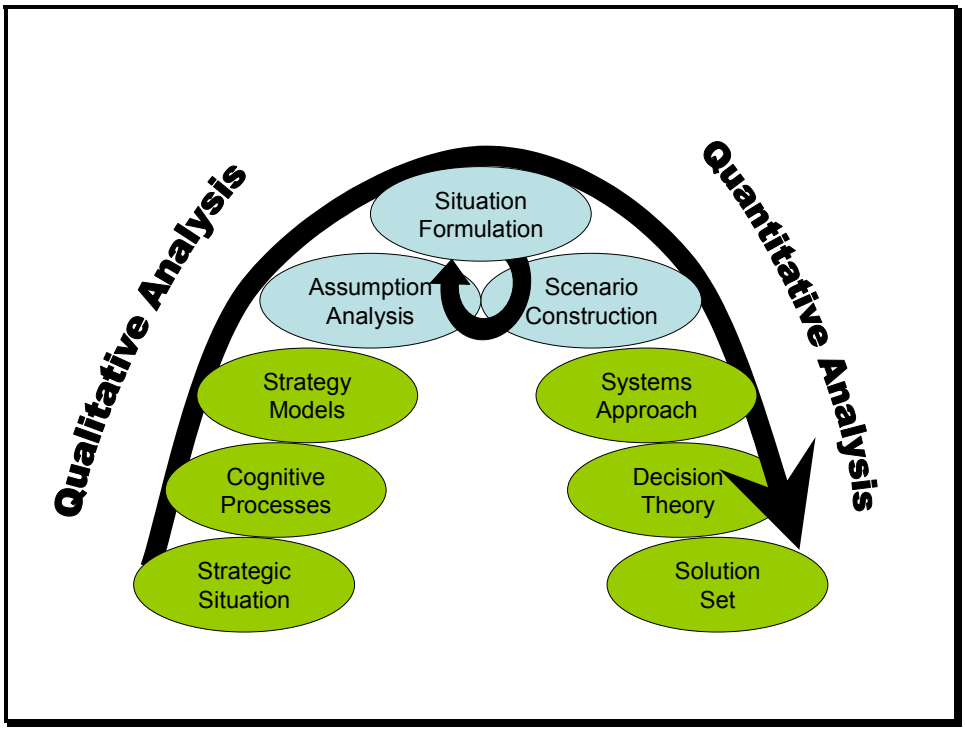


Figure 7. CSM bridge. (Acar, 1987)

On the other hand, a pure process-consulting approach to problem solving focuses primarily on group interactions and discussion to formulate problems and propose solutions, without considering the power of a quantitative approach to analysis that optimizes choice. Formulated solutions and strategies may be less than optimal or may not consider incompatibilities between goals and structural contradictions present in the situation. CSM draws on the strengths of these two different approaches to problem solving. It brings together in one method the benefits of a quantitative modeling approach and the power of a dialectical group process approach. The method overcomes the solution-oriented approach of OR and provides a more rigorous analytical approach to process-oriented methods.

3.2 CSM Semantics

The graphically based CSM method uses causal maps developed by individual managers that model the structural systemic elements of their situation and show how change is propagated through the system.

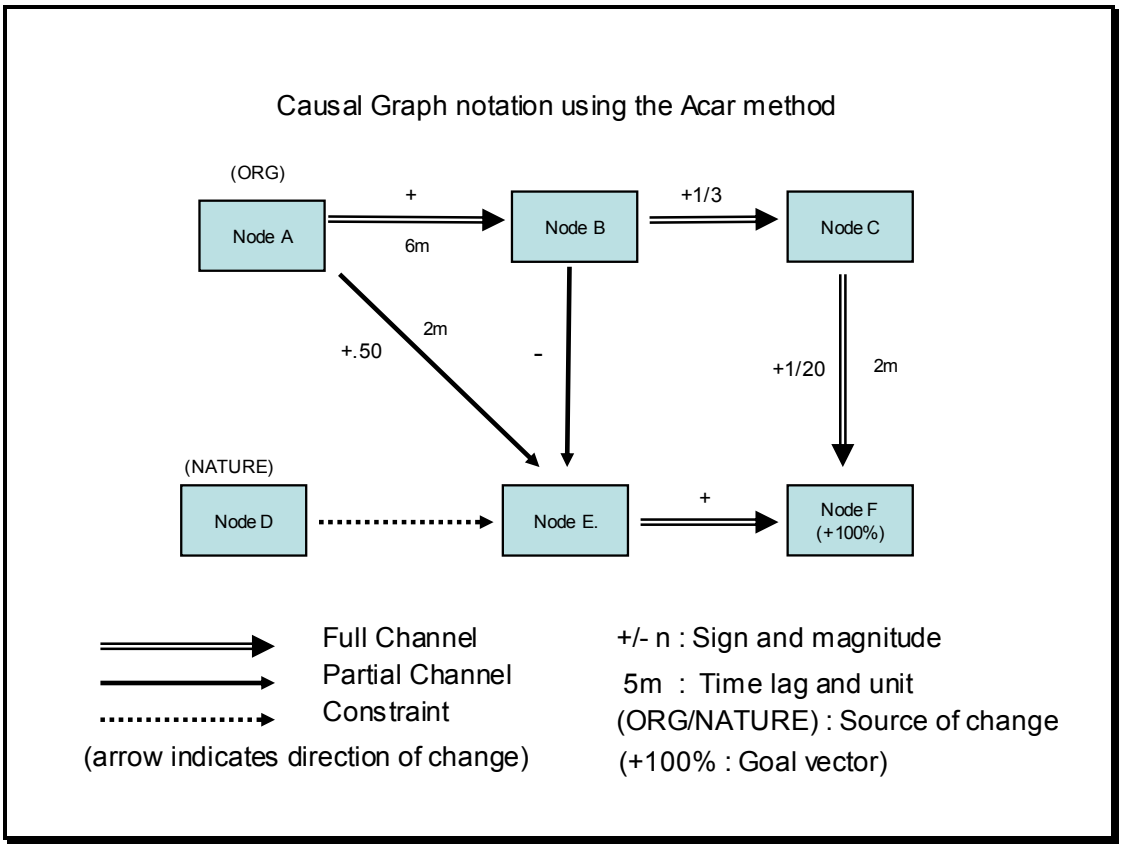


Figure 8. An example of CSM (Acar, 1983)

Looking at Figure 8, elements subject to change are modeled with respect to type of channel (single arrow or *half channel*, double arrow or *full channel*, and dashed arrow or *restricting channel (constraint*), sign and magnitude of change (+.50), and minimum thresholds and time elements (6m = six months). Sources of change are noted by the name of the source in parenthesis and goal vectors are noted on the appropriate node by the sign and percentage of change desired. Full channels transmit change as dictated by the signed magnitude, direction, and time lag parameters. In order for half-channel linkages to activate, all half-channel inputs to a node must be activated. The flow of change through the system is thereby quantified and may be used for a more in-depth analysis. Thus causal maps can become the basis of a simulation study of the strategic landscape which allows for evaluation of strategic choices.

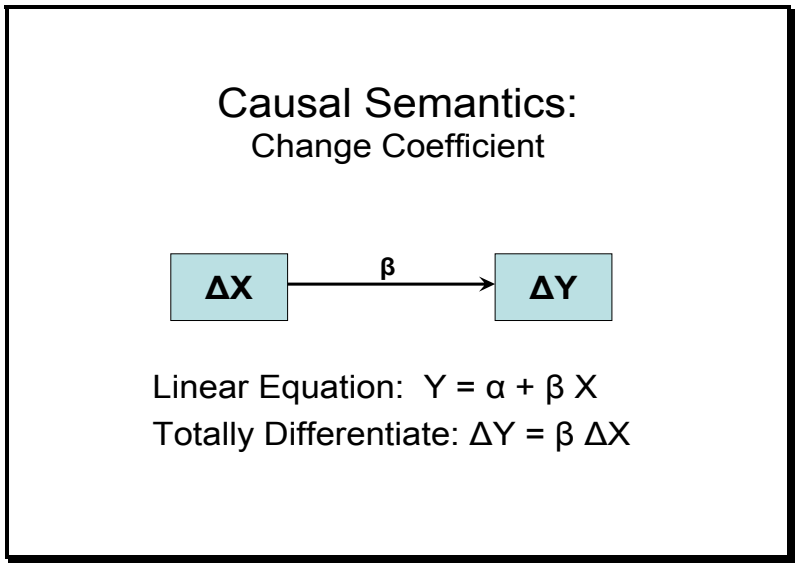


Figure 9. Link coefficient notation - mathematical basis.

Figure 9 shows the underlying mathematical notation for a causal link coefficient of change. The beta coefficient is based on the simple linear equation. Alpha is a constant and changes in Y are equal to the constant plus beta times X. If the equation is totally differentiated, we get delta Y is equal to beta times delta X. This drops the constant and simplifies the notation system. For a detailed explanation see section 2.5.1. Causal Graphing revisited: Recent Developments in Acar (1983).

3.3 CSM Process

As a process-oriented method, Acar's (Acar, 1983) CSM progresses through several stages or levels in the development of causal graphs. The first stage *(Level 0)* is a survey stage that gathers specific information from individual stakeholders on the principal elements (nodes A,B,C, etc. in Figure 8) that constitute the problem situation. The next stage *(Level 1)* is a structuring stage in which the principal elements are graphically related to reveal the qualitative structure of the situation (connecting arrows in Figure 8). The final stage *(Level 2)* adds detail to the graph resulting in a quantitative model of the situation with computational properties (additional notations on the arrows in Figure 8).

3.4 Levels of Analysis

The development of causal maps through successive stages supports three types of analyses: backward analysis, structural analysis and forward analysis (see Figure 10).

3.4.1 Structural analysis

At the Level-0 Survey stage, one takes stock of the factors considered and assesses them for the extreme cases of incompleteness of scope or overlaps. Like other digraph-based approaches, structural analysis in Acar's (Acar, 1983) CSM framework lends itself to the basic structural analyses using the algebra of "adjacency matrices," the theorems and results of which are described in the classic text by Harary et al. (1965). The structural analyses of graph change, reachability, graph scope and connectivity can be conducted at the structuring and graphing stages.

3.4.2 Backward analysis

In addition, CSM-specific analyses that can be performed with causal graphs allow for comparison between individual decision-makers, (or decision-making groups) views of the situation and changes in those views over time. At Level 1, this is the backward analysis of assumptions. Level-1 through Level-2 stages support backward analysis of key assumptions about the problem situation's factors, variables, actors and their relationships, and detailed information on qualifications and quantifications of those relationships (see Figure 8). Backward analysis is accomplished through specification of key actors and factors in a strategic situation. The underlying method and theory of backward analysis is based on Strategic Assumption Surfacing and Testing **(SAST)** (Mason & Mitroff, 1981). Participants are engaged in working backward from preferred strategies to hidden, underlying assumptions through dialectical inquiry. Once surfaced, these formerly tacit assumptions are tested through the process of dialectical inquiry (debate). The surfaced and tested assumptions form the basis for the development of newer robust strategies.

In causal mapping, backward analysis is the process of identifying the key actors and variables in the system of interest and creating a graphic representation in a semantic network map. The process proceeds in stages starting with a divergence phase in which participants create individual causal maps. The resultant maps are personal representations of the system of interest and its environment. This process sets the stage for a convergence phase in which a common

causal map is collaboratively developed using the insights from the individual maps. The process of convergence creates consensus through dialectical inquiry. Major and minor assumptions about the situation are surfaced and tested, resulting in a commonly held perception of the situation. The system is classified as a Singerian-Churchman inquiring system (Churchman, 1971) because different perceptions of a situation are merged through a "sweeping-in" process providing an inter-subjective view of the situation.

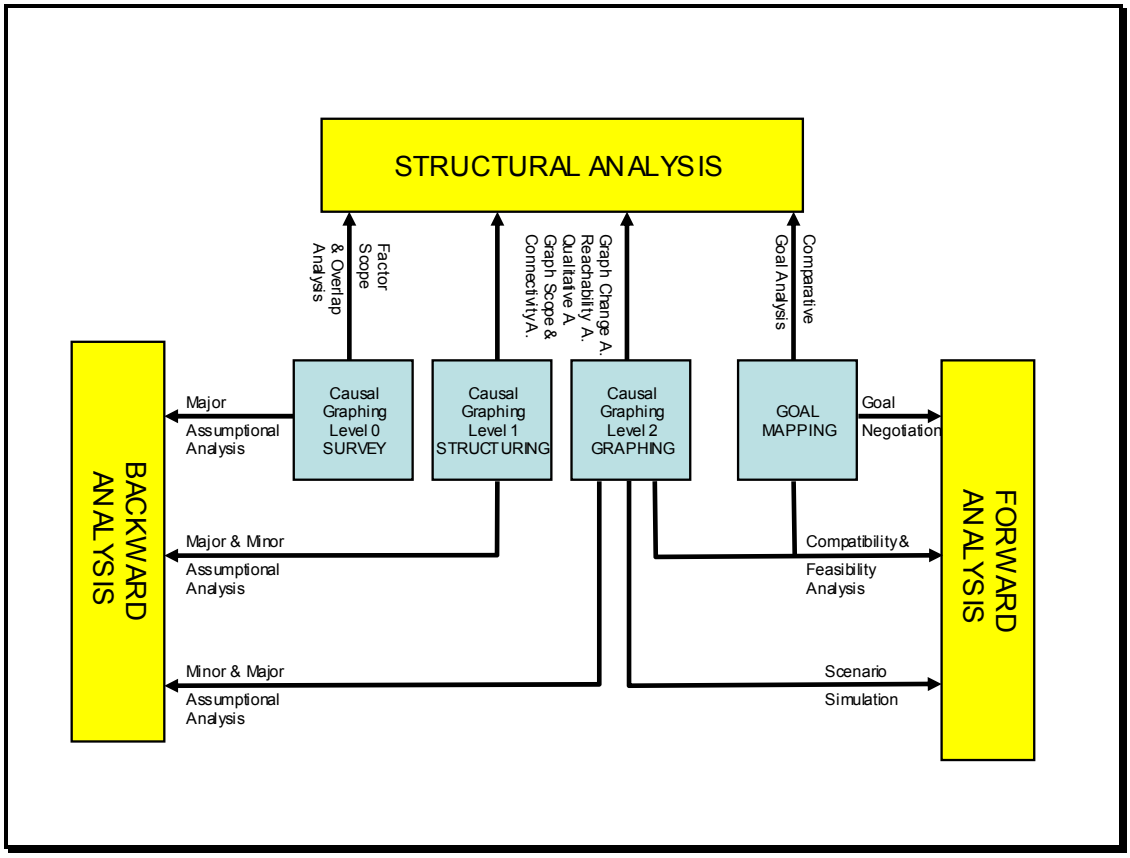


Figure 10. A synoptic view of Acar's system: components of causal mapping. (Acar, 1983)

3.4.3 Forward analysis

Level-2 and goal mapping stages link situation formulation with a forward-analysis capability. *Forward analysis* includes goal negotiation, scenario simulation, and analyses of goal compatibilities and feasibilities.

3.5 Proposed Agent Architecture

Developments in DAI and MAS have renewed interest in the project and offer unique solutions to both problems in a powerful and integrated way. The first limitation suggests the use of a distributed agent architecture for support of CSM. Distributed agent architectures provide a secure, networked, and platform independent infrastructure for building collaborative applications. The second limitation is addressed through agent-based modeling.

AI is touted as a powerful tool for use in support of strategic decision making and in solving unstructured problems. Expert systems solutions promised to harness the knowledge of experts in organizations and apply that knowledge to new problems. Expert systems are not particularly effective at dealing with unstructured problems, however. They require a narrow, well-defined problem space for which heuristics are readily available. This is not the realm of strategy and policy.

A strategic decision support system based on CSM is a complex sociological structure composed of individual managers. Managers are connected through complex knowledge structures that represent the knowledge of each manager and the interrelationships between the causal elements maps expressed as causal maps. Managers also perform roles in value structures that control messages sent to the blackboard. The blackboard is a flexible and adaptable learning structure that represents a multi-leveled conceptual knowledge model. Complex interactions between individual and public value structures serve as guarantors of the system. It is clear that the nature of the problem is a real world complex system that is distributed with multiple loci of control, described as a hierarchy of interacting components, organized for collaborative problem solving, and developed independently with multiple interacting components that incrementally add functionality. Thus it is a suitable design subject for a multi-agent oriented approach.

3.5.1 System analysis

CSM is both a collaborative process and an analytical framework. As a process it can be viewed as a collaborative dialectical conversation that through negotiation develops a common conceptual framework (causal map) of the problem situation implemented as a blackboard structure. As such, it extends the process of “assumptional analysis” described by Churchman (1968) and Mason & Mitroff (1981). In addition, because of its forward analysis capability, the full CSM process allows for goal mapping and development of scenarios that aid strategic planning and formulation.

3.5.2 System design

The type of MAS support is detailed for each of the three main types of analysis available through CSM (see Figure 11).

The fundamental analyses of CSM can be supported by application of agent technologies:

Backward Analysis. In our proposed system, Environmental Scanners are designed to use causal maps as a contextual framework for automated intelligent filtering.

Structural Analysis. Mapping facilitators are designed to support dialogue management for comparison and integration of causal maps.

Forward Analysis. Scenario managers and their analysts are designed to support forward analysis by using the simulation capabilities of causal maps.

3.5.2.1 MAS support for backward analysis

Agents can also be used to support automated backward assumptional analysis that further refines and validates the maps being developed. Using the causal map as a blackboard structure, information sources can be identified in the organization's knowledge bases, model base and database, to support and confirm specific variables and the quantitative assumptions about them. Thus causal maps may be used as a contextual framework for intelligent, directed environmental scans of the corporate external and internal environment. Business Intelligence (BI) applications currently perform this role but lack strategic direction. One of the newest BI tools, Business Activity Monitoring (BAM), provides monitoring of business activities in real time from either a tactical or strategic perspective. BAM will be an increasingly important aspect of Business Intelligence (BI) applications over the next decade (McCoy & Govekar, 2002; Gassman, 2003). A multi-agent inquiring system provides a much needed conceptual framework for directing the information processing found in state of the art business intelligence systems.

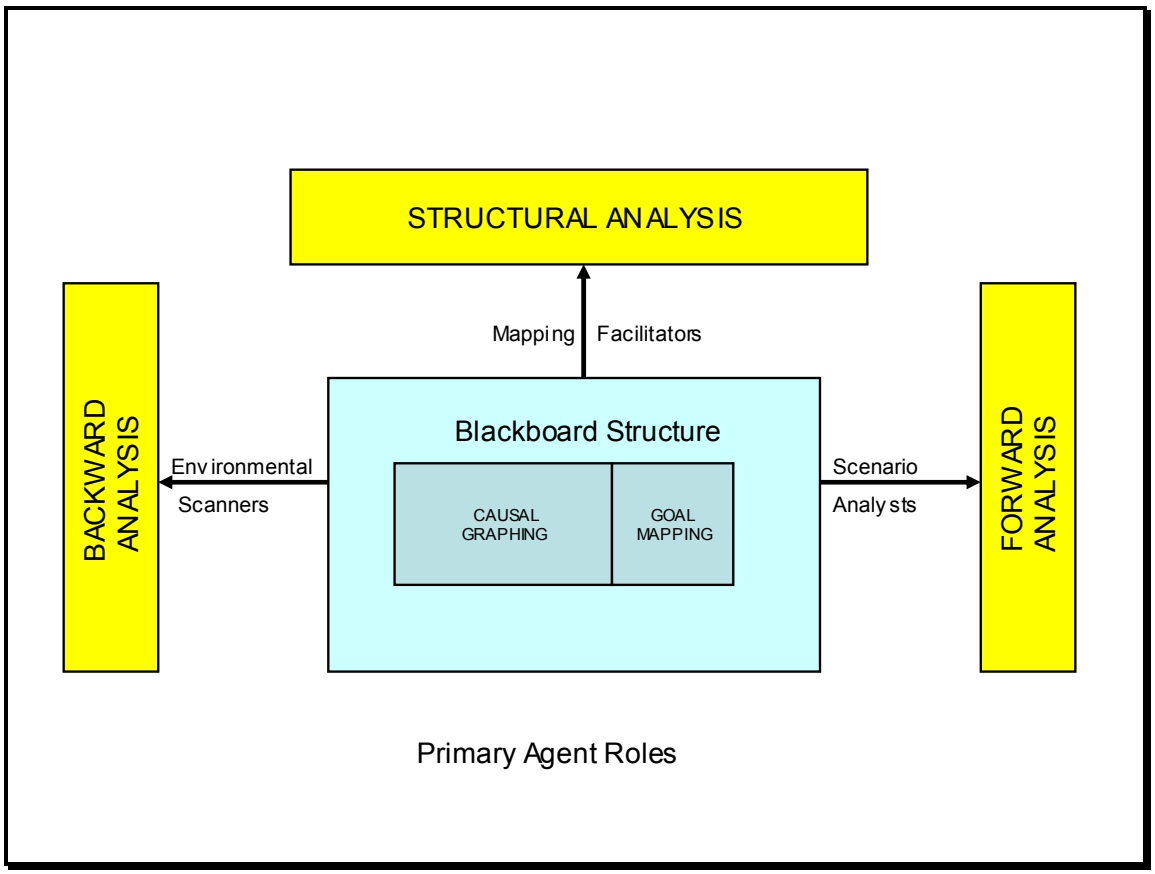


Figure 11. Primary agent roles in the proposed operationalization of Acar's (1983) CSM.

3.5.2.2 MAS support for structural analysis

The key development here is to build agents to analyze the factor and scope overlap of individual maps and to propose potential changes to facilitate merging individual maps into a common map. Over time, agents could use stored graphs and change histories to refine the facilitation process. The pattern of changes and the recurring themes in graphs can be identified by agents, used to help critique maps developed by users, and provide for an intelligent facilitator of the dialectical process of map creation.

Another type of structural analysis provided by CSM is *goal analysis*. Individual goals are specified through goal vectors which represent a final goal state of a causal map. Both qualitative Boolean analysis and quantitative comparative analysis of individual goals are supported. Agents would be an ideal method for performing this type of analysis, thus facilitating

the comparisons and negotiation of individual goals into a common vision of the ideal state of affairs for the problem situation. This is actually part of the forward analysis capability of the system which is addressed in the next section.

3.5.2.3 MAS support for forward analysis

The development of strategy and competitive advantage has been the subject of a large body of research in strategic management. Recently multi-agent systems using genetic algorithms have been used for testing and developing strategic theory (Phelan, 1997). The central focus of Phelan's research was a simulation method using an adaptive MAS. A learner-classifier (Holland, 1992) machine learning simulation based on genetic algorithms was employed. Phelan's simulation used agents to simulate strategic situations that would be difficult to study experimentally (as is the case with strategy.) In Phelan's research, agent systems are used to simulate the strategic landscape or topology and the strategies pursued to optimize performance in a dynamic environment. The research showed the benefits to be gained from creating a simulation test bed for testing generic strategies, but was not linked to specific environmental topologies and was limited to a micro-world testing of strategic theory.

The prototype developed in this dissertation is poised to go much further. A multi-agent causal map implements a specific organizational strategic landscape which is then used to test development of robust adaptive strategies for specific organizational situations. CSM forward analysis supports goal negotiation feasibility and compatibility analysis. Individual goals can be identified and analyzed for their compatibility with the problem situation by running a causal map forward in time to determine the feasibility of the desired goal states. Desired goal states may not be compatible with the situation as conceived. Agents here could be used to evaluate goals based on a forward analysis of the map, thus providing feedback on the feasibility of individual or negotiated goals. Tactics can be modeled as inputs to the controllable variables in the system. Groups of tactics are modeled as different scenarios that allow for the development of strategic themes at work under multiple scenarios. Agent-based modeling and simulation potentially offers new solutions to dealing with the complexity and uncertainty found in causal maps. The problem with ABM modeling tools in general is the same, though, as other computerized approaches to modeling and simulation: they require substantial programming expertise and are expensive to develop and maintain.

The development of a user friendly ABM approach to the simulation of strategic scenarios would be unique in that it addresses this fundamental problem. A marriage of powerful

distributed artificial intelligence techniques based on complexity science with a well defined method for problem formulation that is embedded in a user friendly modeling interface would be a welcome tool for strategy making. Such a tool should be designed to allow decision makers to collaboratively develop and simulate strategic situations quickly and inexpensively. This problem is specifically addressed next and is explored in the balance of this work.

CHAPTER 4 PROTOTYPE DESIGN

In this chapter an object-oriented, user-friendly system for editing of causal maps is linked with an agent-based modeling and simulation system. Developed maps can then be put into motion (simulated) using agent-based modeling. Agent-based modeling is uniquely suited to analyzing the complex loops found in causal maps. The complexity such loops embody is analytically intractable and precludes the use of a strictly object oriented approach to design. The simulation capability provided by agent-based modeling allows users to examine the forward implications of a causal map facilitating the understanding of frequently counter-intuitive situations.

4.1 The Problem of Automated Support for Causal Mapping

The benefits of automated support for cognitive mapping are well established and researched (Eden & Ackermann, 1998; Chaib-draa, 2002; Heintz & Acar, 1994). Strategists, however, have long been suspicious of computer modeling and simulation as a support despite the availability of powerful tools used by management science and operations research. These tools are not user friendly to decision makers, are time consuming and expensive to develop and maintain, and frequently result in models that fit the designer's conception of the problem rather than the decision-maker's (Mason & Mitroff, 1981).

The scope and complexity of the CSM system have dictated a partitioned approach to development. An early prototype of the system (Heintz & Acar, 1994) developed an object model that supported causal mapping semantics and process. While the prototype (implemented as a Smalltalk application) demonstrated the ease of development with object oriented techniques, it was limited to the graphic editing of causal maps. This initial research identified several technical problems associated with the implementation that represented limitations of the prototype.

4.1.1 The need for a networked, platform independent application.

The advent of internet technologies, client-server techniques and more powerful cross platform development tools such as Java in the late 1990's offered a potential solution to the problem of networked platform-independence. An early test system was constructed for collaborative editing and user management in a networked environment using Java's remote method invocation (RMI) capabilities. Building a secure multi platform infrastructure to manage users, sessions, security, and concurrent editing of causal maps is a substantial effort using Java's foundation classes alone. This Java-based pilot implementation pointed the way to solving this problem, but the custom built platform was not easily integrated with other independently developed systems. A second major limitation remained elusive:

4.1.2 The need for simulation and forward analysis of causal maps.

The present development goes further in this regard than the object-oriented approach of Heintz & Acar (1994). The complexity of causal loops in the models and calculations of successive waves of change through the model when simulated presented a substantial design challenge for developing automated scenario support using object-oriented techniques alone. The initial prototype study suggested an AI solution. However, no specific implementation was suggested. This constituted a serious limitation of the prototype and precluded its use for scenario development.

4.2 Prototype Development Platforms

The outcome of the design process is to produce an intermediate level abstraction that can then be implemented (prototyped) through more traditional techniques. Once a working prototype is developed, experimentation can take place and lead to refinements in the design. Because of the emergent behavior of agent systems, an experimental approach with prototyping is required to gain experience with the actual effects of the system in a production environment. In general, simulation is a useful research tool for research and has been successfully used as a research technique in strategy research as well as other disciplines (Phelan, 1997). The advantage of using MAS for development is that a simulated system environment can be created using MAS to test how the system works in a variety of test situations which can be systematically controlled.

A simulation can be used to gain experience with the emergent behavior of the system and the characteristics of the system interactions with users.

The prototype explored in this research uses agent-based modeling and simulation to develop the forward analysis capabilities of CSM. Two agent-based modeling tools were investigated for potential use as an implementation environment for CSM: Swarm and Repast 2.0.

4.2.1.1 Swarm

Swarm (Swarm Development Group, 2003) is a software package for the multi-agent simulation of complex systems, originally developed at the Santa Fe Institute. Developed as a research tool for a variety of disciplines, Swarm is a simulation system using collections of concurrently interacting agents. Researchers have used Swarm to implement a large variety of agent-based models. The Swarm Development Group (SDG), a not-for-profit organization, supports the development of the Swarm Simulation System and its use by a community of researchers.

Swarm is freely distributed for use under public license as a set of libraries that facilitate implementation of agent-based models. The conceptual basis of Swarm comes from the field of Artificial Life. Artificial Life is a branch of artificial intelligence that studies biological systems and, through abstract modeling of biological mechanisms such as adaptation, identifies dynamic characteristics that become the basis of artificial modeling.

Since the task of engineering a multi-agent simulation system is beyond the capability of most scientific researchers, the Swarm project was initiated in 1994 at the Santa Fe Institute in New Mexico. First available in 1996, the swarm library has been used by researchers in biology, anthropology, computer science, defense, ecology, economics, geography, industry, political science, and management (Daniels, 1999).

4.2.1.2 RePast2.0

RePast 2.0 (*REcursive Porous Agent Simulation Toolkit*) (University of Chicago: Social Science Research Computing, 2002) is another swarm-like agent-based simulation and modeling tool. RePast is a software framework for creating agent-based simulations using the Java language. It provides a library of classes for creating, running, displaying, and collecting data from an agent-based simulation.

> RePast envisions a simulation as a state machine whose state is constituted by the collective states of all its components. These components can be divided up into infrastructure and representation. The infrastructure is the various mechanisms that run the simulation, display and collect data and so forth. The representation is what

> the simulation modeler constructs, the simulation model itself. The state of the infrastructure is then the state of the display, the state of the data collection objects etc. The state of the representation is the state of what is being modeled, the current values of all the agents' variables, the current value of the space or spaces in which they operate, as well as the state of any other representation objects (e.g. aggregate quasi-independent "institution" objects). The history of the simulation as a software phenomenon is the history of both these states, while the history of the simulation as a simulation is the history of the representational states (University of Chicago: Social Science Research Computing, 2002).

Similar to Swarm, RePast allows a user to build a simulation as a state machine in which all the changes to the state machine occur through a schedule. RePast moves beyond the representation of agents as discrete, self-contained entities in viewing agents as social actors who are "permeable, interleaved and mutually defining, with cascading and recombinant motives. Repast supports the modeling of belief systems, agents, organizations and institutions as recursive social constructions" (University of Chicago: Social Science Research Computing, 2002).

RePast is therefore more applicable to the modeling of social networks and interactions that are commonly found in business and management environments and particularly in the semantics of causal mapping. Simulation support for network topologies is an important distinguishing feature between RePast and Swarm. In Swarm environmental topologies are represented as a two- or three-dimensional grid or torus,[3] which is a natural way to view biological or ecological systems. Social science environments, including business management applications, are more naturally represented through network topologies, making RePast the logical choice to implement causal maps (given the node/link semantics of the method.)

4.3 User Interface Design

The user interface is programmed using Java's Swing interface components. This library of graphic interface objects provides an easy to define user interface with standard components that allow the developer to follow the common user interface (CUI) guidelines. CUI guidelines provide for consistent interfaces that minimize user learning curves and give the application an "intuitive" feel. Bar menus, popup menus, dialogue boxes, etc. all provide standard windowed interface components. In designing the user interface, previous developments were followed. An early prototype (Acar, 2000) implemented a Java based collaborative tool that provided a network

based application for sharing and editing maps. This prototype was on the development of a client server application with minimal user interface development and no support for forward analysis of causal maps.

The user interface components (see Figure 12) are based on drop-down menus that follow CUI specifications. For example the "File Menu" is the first menu item on the bar menu at the top of the application. The File menu allows the user to create "New" maps, "Open" existing maps, "Close" the current map, "Save" the current map (See panel A).

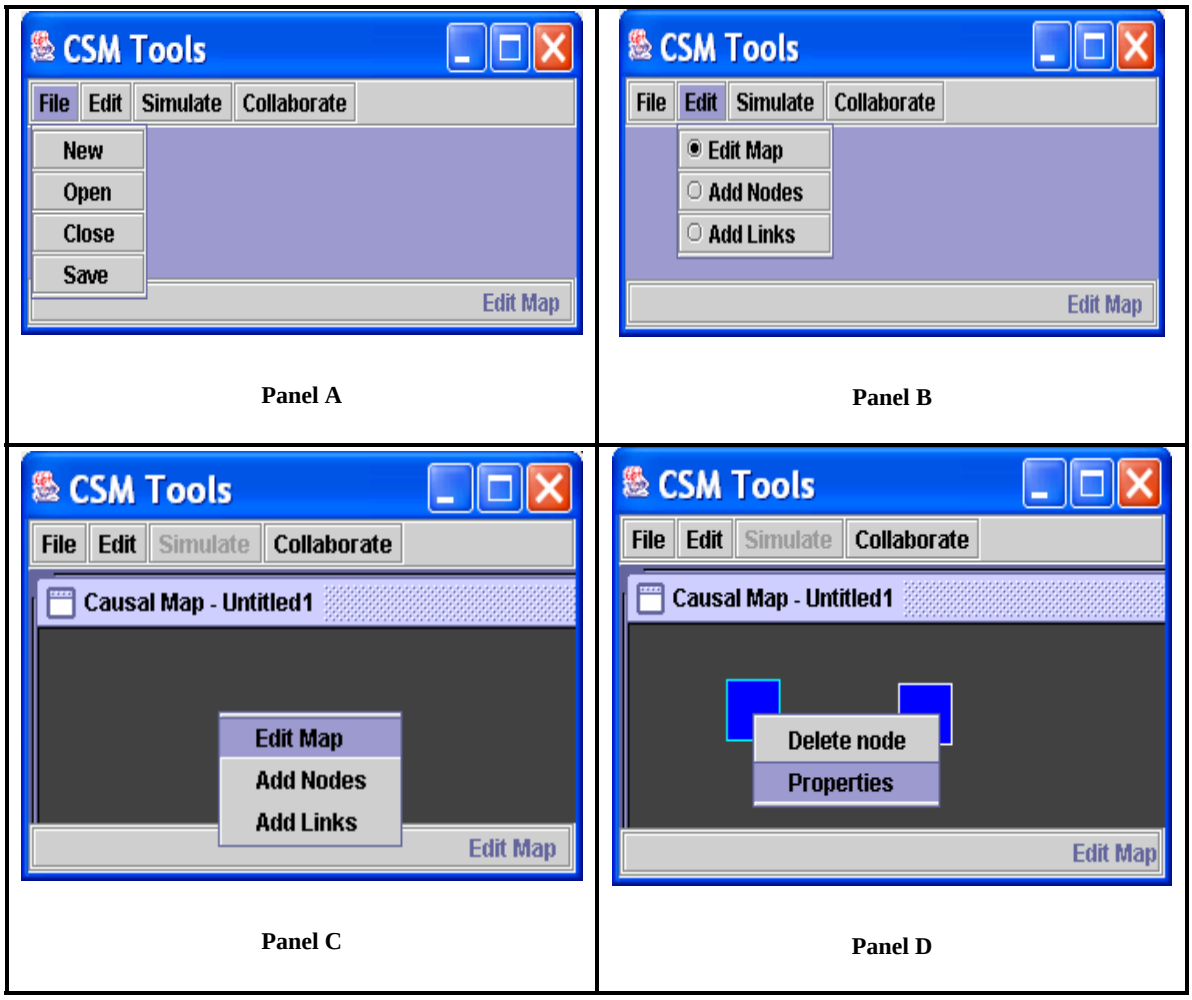


Figure 12. User interface components

[3] A micro-world environment that is basically a two dimensional grid can "wrap" in any direction. For example if agents wander off the left edge of the world they appear on the right hand side of the grid. This is a torus implementation.

The “Edit” menu allows the user to Edit the map, Add Nodes, and Add Links (see Figure 12, panel B). In the “Edit map mode” the user may select and reposition nodes and links on the map. When “Add Nodes” is selected the user can add nodes with each left mouse click. When “Add Links” is selected, the user can add links between nodes by left clicking the first node and then dragging the link to the receiving node. This method of adding nodes and links allows a user to work efficiently by first adding nodes, then adding the links between them. When a new map is created, a map drawing surface is created with the name “Untitled1.” Right clicking on the grey drawing surface will present a pop-up menu with the same options as the “Edit” menu (see Figure 12, panel C). The “Edit” menu allows the user to switch between “edit mode” where nodes and links can be repositioned, deleted, or edited, and “add mode” where nodes and links may be added to the drawing surface but not edited. This is an efficient way to draw maps since the user is not required to continually switch between add and edit mode for each node or link added, as is common with most drawing interfaces. One key aspect of the usability testing performed in the research was to test this design decision since it represents a potential problem for users who are not used to doing things this way. In “Edit Map” mode (indicated on the status bar on the lower right corner) right clicking on a node allows the user to edit the properties of a node or to delete the node (see Figure 12, panel D).

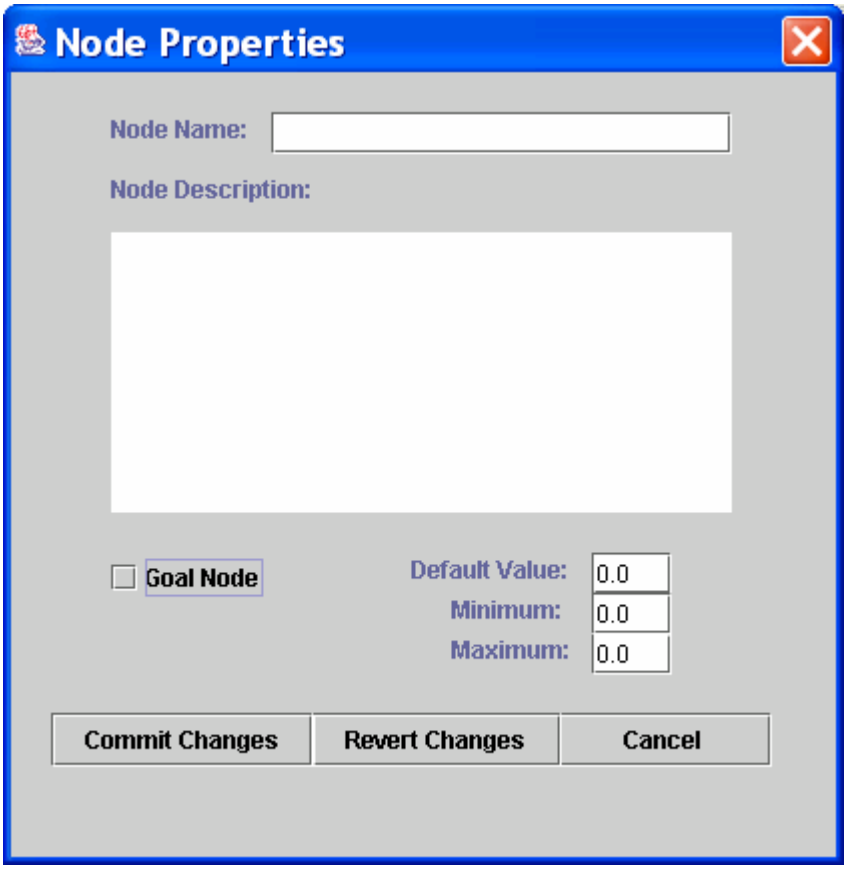


Figure 13. Node property sheet.

Choosing the "Properties" option allows the user to edit the node/link property sheets. Figure 13 shows the node property sheet. Fields are available to enter the node name, description, default value, minimum and maximum threshold values, and a check box that indicates whether the node is a goal node. If the goal node property is checked, it's value is automatically graphed in a simulation allowing the user to evaluate goal expected values. Command buttons allow changes to be saved (Commit Changes), discarded (Revert Changes) or discarded changes and close the property sheet (Cancel).

Right clicking a link allows the user to edit the link property sheet or delete the link. The link property sheet (see Figure 14) pops up when the user chooses "Properties." Fields are available to enter the link name, description, change coefficient, time lag, and type of channel. Buttons allow changes to be saved (Commit Changes), discarded (Revert Changes), or discarded changes and close the property sheet (Cancel).

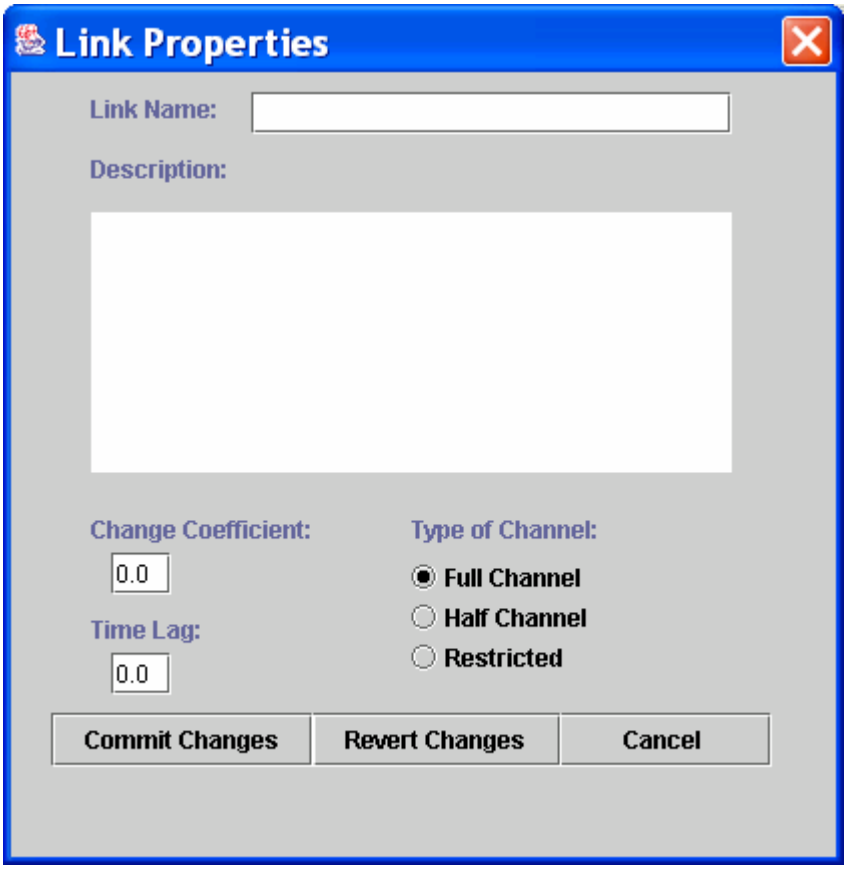


Figure 14. Link property sheet.

The "Simulate" menu allows you to simulate the map with the "Run Simulation" item. The "Parameters" menu item is not currently used. When the "Run Simulation" item is chosen the simulation tool bar is displayed (see Figure 15).

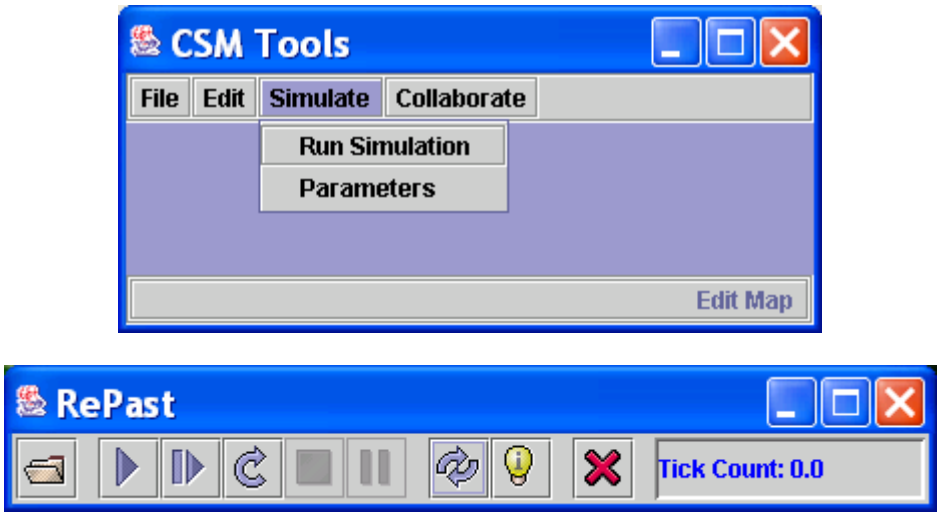


Figure 15. Simulation controls.

The primary icons used to control a simulation are:

 The play button, which "plays" the simulation until it stops.

 The step button, which steps through the simulation one time tick at a time. The status display on the right shows the current tick count of the next time step to be executed.

 The stop button, which stops the simulation if it has been played and is not yet finished.

 The pause button, which pauses the simulation; it then can be started again with the play button.

 The reset button, which resets the simulation, reinitializes all variables, and allows the simulation to be started .

 The cancel button, which exits the simulation function.

4.4 Agent-based Simulation of Causal Maps

The *power of this application is in developing specific environmental topologies* using a causal mapping approach and then simulating the strategic landscape with an agent-based modeling and simulation tool. This simulation system is integrated into the editing interface of the application. Forward analysis (simulation) of maps can be done at any time to test the implications of created maps. RePast provides a robust simulation environment and the linked editing interface provides an easy to use graphic model building tool for RePast simulations.

In order to illustrate this, two hypothetical maps were constructed, one representing a possible terrorist's conceptualization of the dynamics of tourist travel to Europe and the impact of terrorist threats on flight availability, air fares, and consumer demand for travel to European destinations. Figure 16 shows such a hypothetical scenario developed for demonstration purposes. An alternate conceptualization of the same situation (see Figure 20) was constructed for comparison purposes. It is that of an economist employing standard supply and demand economic theory to the same situation. The underlying assumptions differ only in the directionality of one of the causal links as explained below. The simulations illustrate how a simple change of assumptions radically alters the implications of the map. These implications are difficult to anticipate just from looking at the maps, but are dramatically revealed when simulated with the software.

In the first example map (Figure 16), increased terrorist threats impact both consumer demand and flight availability as flights are canceled. The link notation shows that increased terrorist threats cause a decrease of 5% in available flights and a decrease of 25% in consumer demand for travel to Europe. A time lag of one week is indicated for both of these links. Fewer available flights forces fares higher which also affects consumer demand. Decreased consumer demand means fewer people travel to Europe. Fewer people traveling to Europe makes threats to air travel less attractive to terrorists, decreasing the threat level. This represents a terrorist's possible conceptualization of the situation with a goal of disruption to a key economic sector in the European economy.

A scenario is constructed around this situation with an increase of 10% in terrorist threats being the only trigger (see Figure 16). This is accomplished by changing the default value in the property sheet of the terrorist threat node to ten. Once this change is committed, the color of the node is changed to yellow to represent a change in the node from its status quo level of zero.

Three goal nodes are also chosen for this scenario. This is accomplished by editing the property sheet for air fares, European tourism, and consumer demand. When the scenario is simulated, a graph (see Figure 19) showing the status of each node is displayed. The graph is automatically scaled as the simulation progresses with the X axis representing simulation time and the Y axis representing the change in the node (either positive or negative) from the initial status quo level of 0.

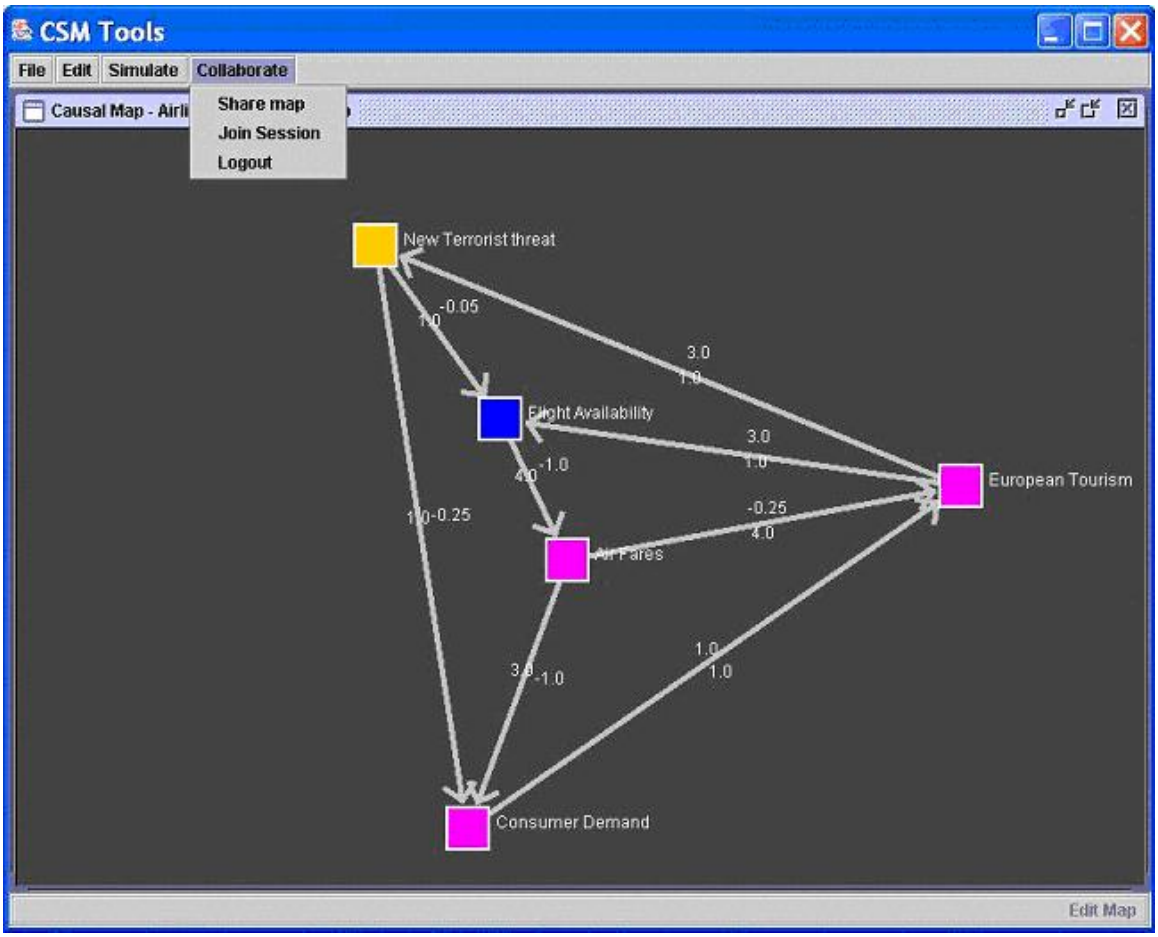


Figure 16. Editing interface

The link coefficients and time lags indicated on the map are only for illustration and are not based on empirical studies. Empirical validation of simulation models is an outcome of the process. At an early stage of map development, the best estimates of decision participants are used to construct individual models. A primary future thrust of this research is to enable agent

based validation of quantifiable assumptions in the model through data mining or other data analytical methods.

Once the scenario is constructed, an agent-based simulation is run. This is initiated by selecting the "run simulation" option from the top drop-down menu (see Figure 15). The causal nodes, now implemented as agents, transmit changes to other agents through the defined linkages.

Simulation time/Variable	Terrorist Threat	Flight availability	Air fares	Consumer Demand	European Tourism
Tick 0	10%	0%	0%	0%	0%
Tick 1	10%	-.5%	0%	-2.5%	0%
Tick 2-4	10%	-.5%	0%	-2.5%	-2.5%
Tick 5	7.5%	-3%	-.5%	-2.5%	-2.5%

Table 1. Simulation of scenario through first five time ticks.

Table 1 shows the values of the nodes through the first five time ticks with values representing the cumulative percent of change from the status quo. Beyond this point calculations become quite tedious and difficult to keep track of due to the differing time lags associated with each link. At tick zero, only the terrorist threat node has changed with the initial trigger value. All other nodes are at their status quo level of 0. As the simulation progress to time tick one, both the flight availability node and consumer demand nodes have a new value. Flight availability has decreased to -0.5%. This value is obtained by multiplying the change in terrorist threat level (10%) times the link coefficient of -0.05 giving the new value. As the simulation progresses, the changes cascade through the map based on the time lags on the individual links. By tick five, the 2.5% decrease in European tourism that occurred at tick two has reduced the terrorist threat level by the same amount to 7.5%. Flight availability is now reduced 3%. Consumer demand and European tourism are unchanged for now. The changes continue to cascade through the system with increasingly wider swings in the node values, sometimes positive, sometimes negative. These cascading changes are graphically represented in Figure 17 and Figure 18.

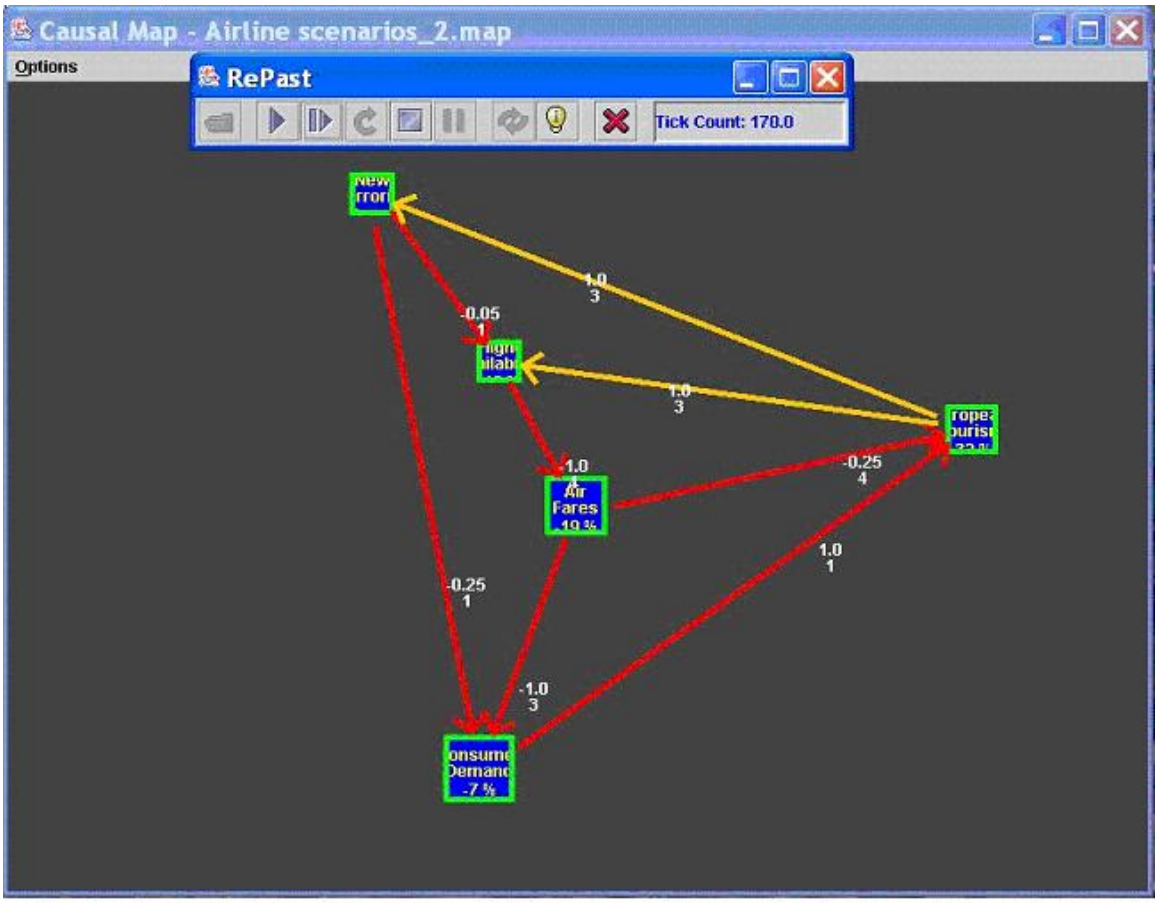


Figure 17. Repast simulation of edited map – simulation time 170 weeks

Emergent patterns of node interaction and system dynamics are viewed and analyzed through animations, custom charts and graphs, and other functionality provided by RePast (see Figure 19). This represents an important advance over traditional causal mapping methods which are static representations of situational dynamics. The ability to easily animate maps and set their dynamics visually into motion is a critical development that makes causal mapping an engaging, dynamic, and interactive support tool for strategic decision support.

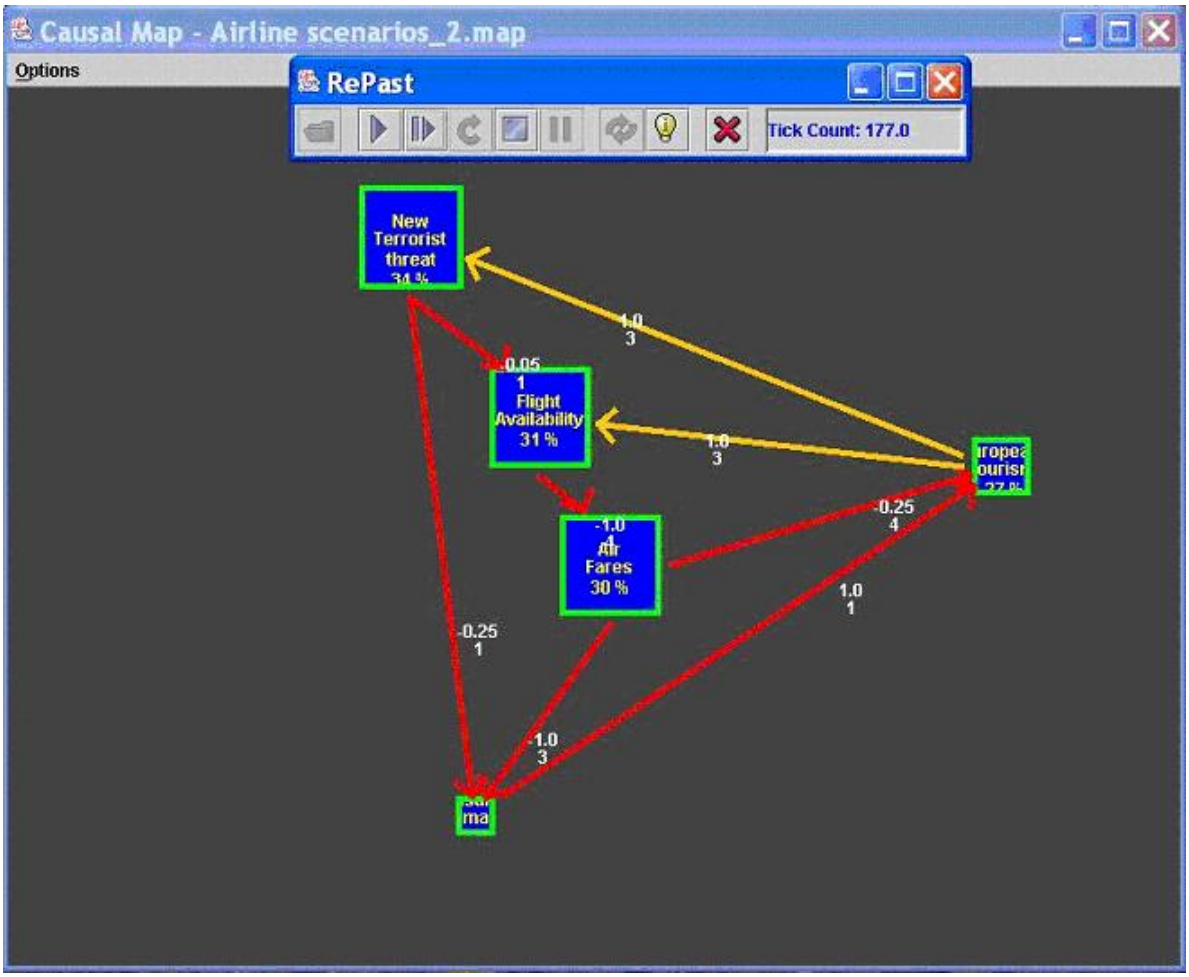


Figure 18. Repast simulation of edited map – simulation time 177 weeks

The propagation of change through the network is color coded to enable visualization. Figure 17 shows the terrorist scenario at 170 weeks of simulated time. The initial point of change in this scenario was an initial increase of 10% over the status quo of terrorist threats against air travel to Europe. As the change propagates through the system, links are colored orange (light colored links) when changes are loaded onto the link and red (dark colored links) when they activate. At 170 weeks, terrorist threats have subsided, consumer demand for travel is on the rise and air fares have fallen. This can be compared with Figure 18, which is the map at 177 weeks of simulated time. Terrorist threats have once again increased as available flights were increased due to increased demand. More available flights provide richer targets of opportunity, etc.

The nodes are either visibly increased or reduced in size and the current change from status quo is indicated inside the node as a percentage. This scenario is unstable; viewing Figure 19 one can see a graph depicting the levels of the various nodes as the simulation progresses through time. The swings between positive and negative values get larger and larger as change reverberates through the system. This of course might represent the hope of terrorism: destabilization of an economic system and constant disruption and economic disaster.

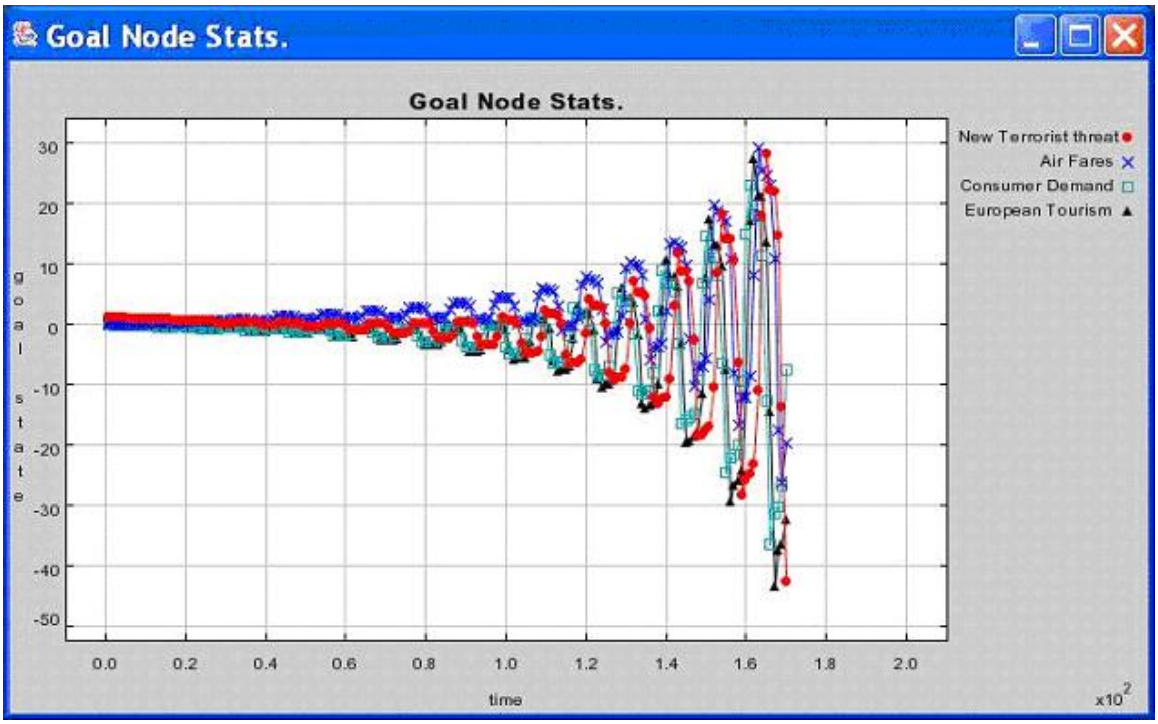


Figure 19. Repast simulation of edited map – node graph

Users examine simulation output to help refine assumptions and check the face validity of the model. The simulated outputs are the implications of the situation as it is currently modeled. An iterative process of backwards assumption modeling and forward simulation analysis allows decision makers to understand the nature of the strategic situation and problem at hand. Alternate scenarios (changes in the initial conditions or changes in map structure) can be created and stored as separate maps.

An economist using economic theory as a guide would have a slightly different view of the situation. Based on the theory of supply and demand, an economist would reverse the directionality of the link between air fares and consumer demand. Flight availability represents supply and consumer demand is demand. Air fares are prices set by supply and demand. Supply

decreases should be offset by decreasing demand, resulting in an equilibrium at a new price level. Figure 20 represents the economist's scenario. The trigger node is still a change in the terrorist threat level and air fares are selected as the goal node. An economist would predict that the system should reach an equilibrium at some new price level.

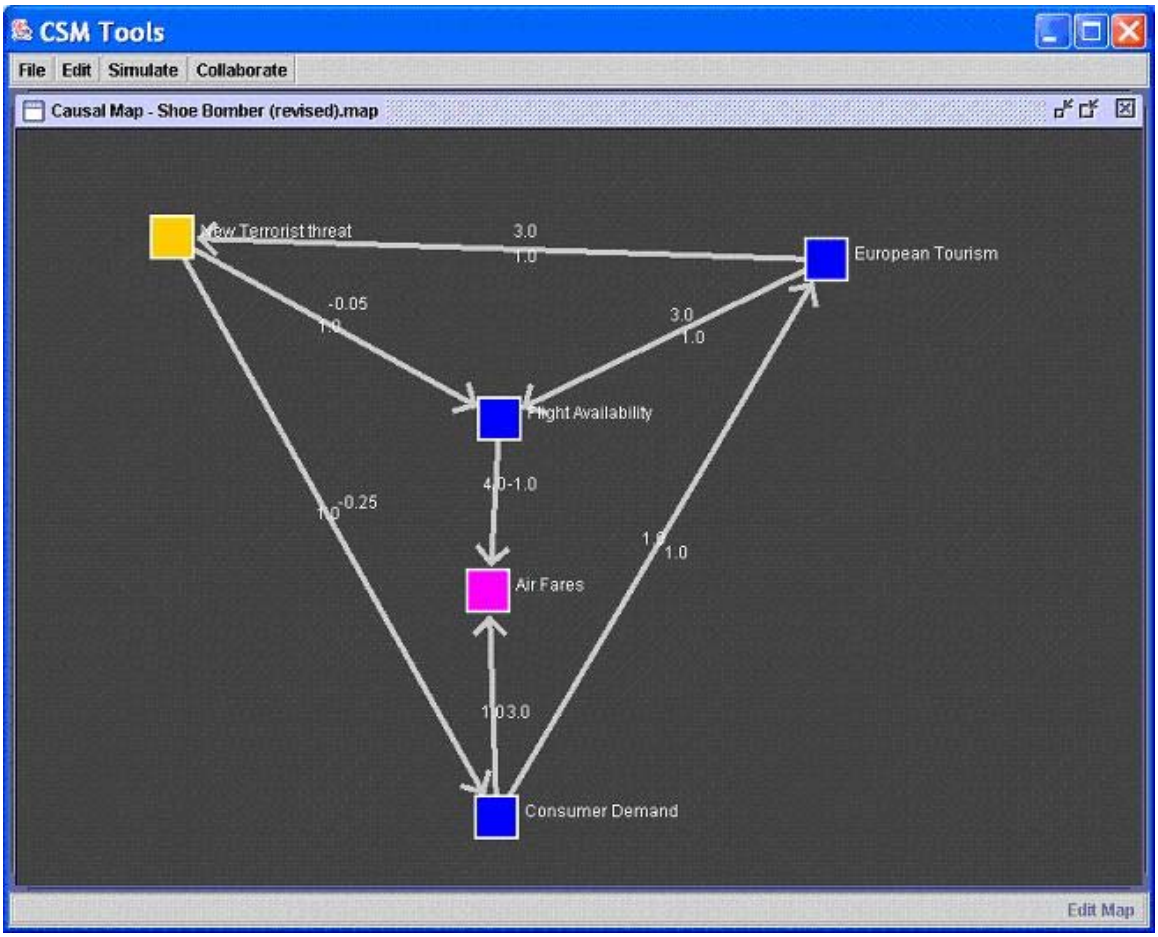


Figure 20. Revised example according to economic theory.

Figure 21 shows the end point for this scenario at time tick 22. The system has stabilized with air fares at a new equilibrium price level that is 0.375% higher than the initial status quo. The implications of this simple change are quite dramatic in that the system no longer exhibits a chaotic pattern and is quite stable.

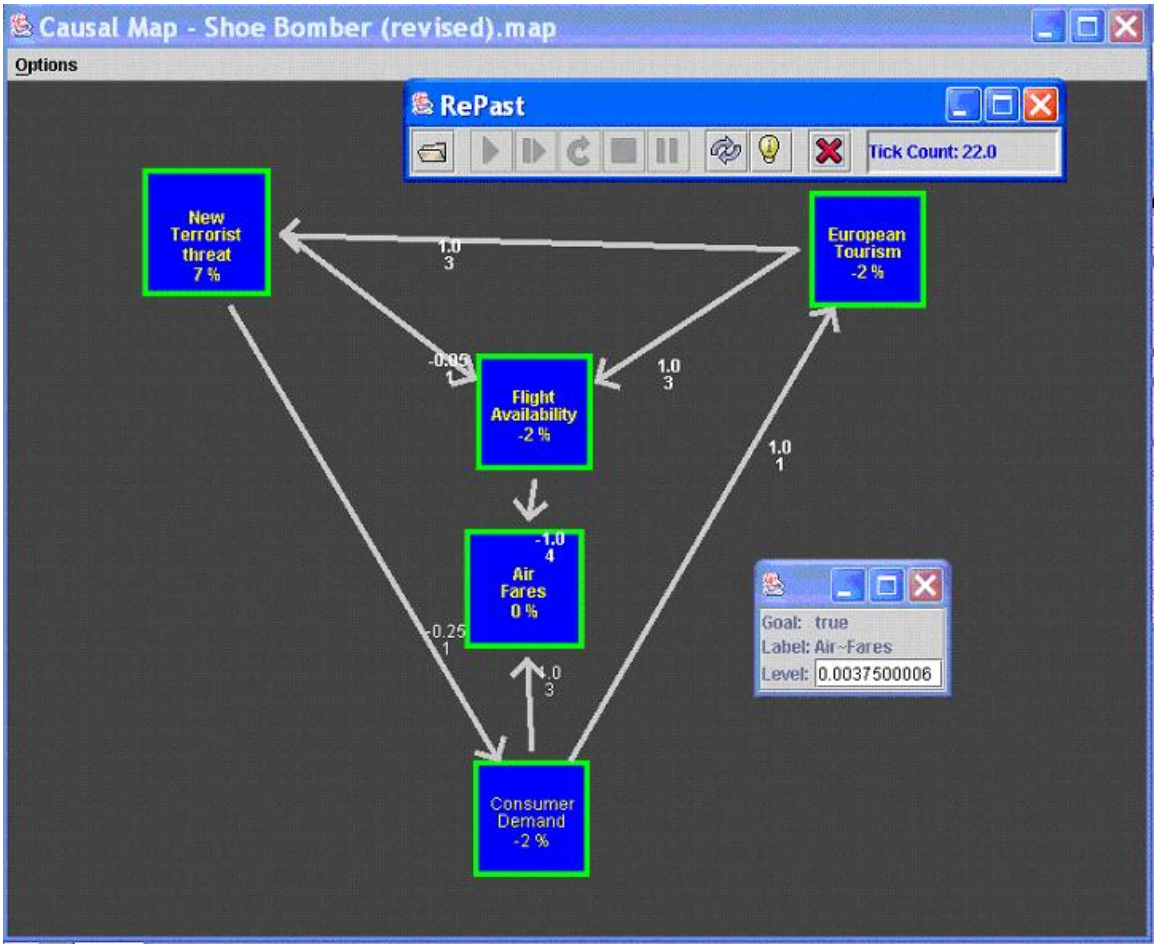


Figure 21. Economist scenario at end of simulation.

4.4.1 Dynamic model construction and agent behavior

The graphic modeling interface facilitates an interaction between modeling of assumptions and simulation of the implications. Mapping models are easily updated, saved, and shared. When a map is simulated, a RePast model is dynamically constructed from the graphically edited map with individual nodes represented as agents. Links between agents constitute the environment of the system, analogous to a communication network.

Agents (nodes) listen on incoming links for changes from associated agents. When changes "arrive," after the time lag indicated on each link, each agent updates its state by adding the new change value times the change coefficient on the link to the status quo level of the agent. Changes are thus cumulative and, since link coefficients can be negative or positive, the status quo level of the agent will increase or decrease. Changes do not automatically affect the agent,

however. Threshold values are defined that allow the agent to block or drop changes from propagating through the system. Received changes are then transmitted on outgoing communication links. Thus agents are both interactive and autonomous.

Links are implemented with a vector structure since successive waves of change can be present on a link at the same time due to the time lags and loops of specific models. As changes mature with the progression of time, they become "Hot" and activate the link. The attached agent then gets the change and updates its state. The change element is deleted from the link and the agent processes the change. Links are not implemented as agents.

The different kinds of links also affect agent behavior. Agents will process *full channel* linkages if threshold values allow. *Partial channels,* on the other hand, are only processed if all incoming *partial channels* are activated. Agents hold activated link values in memory and when all *partial links* are active the change is processed. *Partial link* processing is not additive in the sense that the change values on each link are added together. These links represent necessary *and* sufficient conditions for a change to take place. *Partial link* changes take the minimum value of the associated links. *Restriction channels* represent qualitative factors that operate as a binary switch on the node restricting or enabling an agent. If a *restriction link* is activated, the associated agent is blocked from processing changes.

4.4.2 Scheduling

Change is initiated in the system by the user's specifying an initial level, either positive or negative, that is different from the status quo level. These changes in initial conditions are made in nodes that represent either external triggers (external agents that affect the system) and internal levers (internal agents that represent controlled agents in the system of interest.) These nodes are graphically represented in the system by a change in the node color to orange. Scenarios are run by changing the initial level of trigger or lever nodes that represent different sets of initial conditions. Primary agent behavior during each time step of simulation time is recursively scheduled.

Agents must continue to process changes that cascade through the system until no new changes are present on links. This means that agent behavior must be continuous. Infinite causal loops can develop as a result. A basic test problem is illustrated in Figure 22.

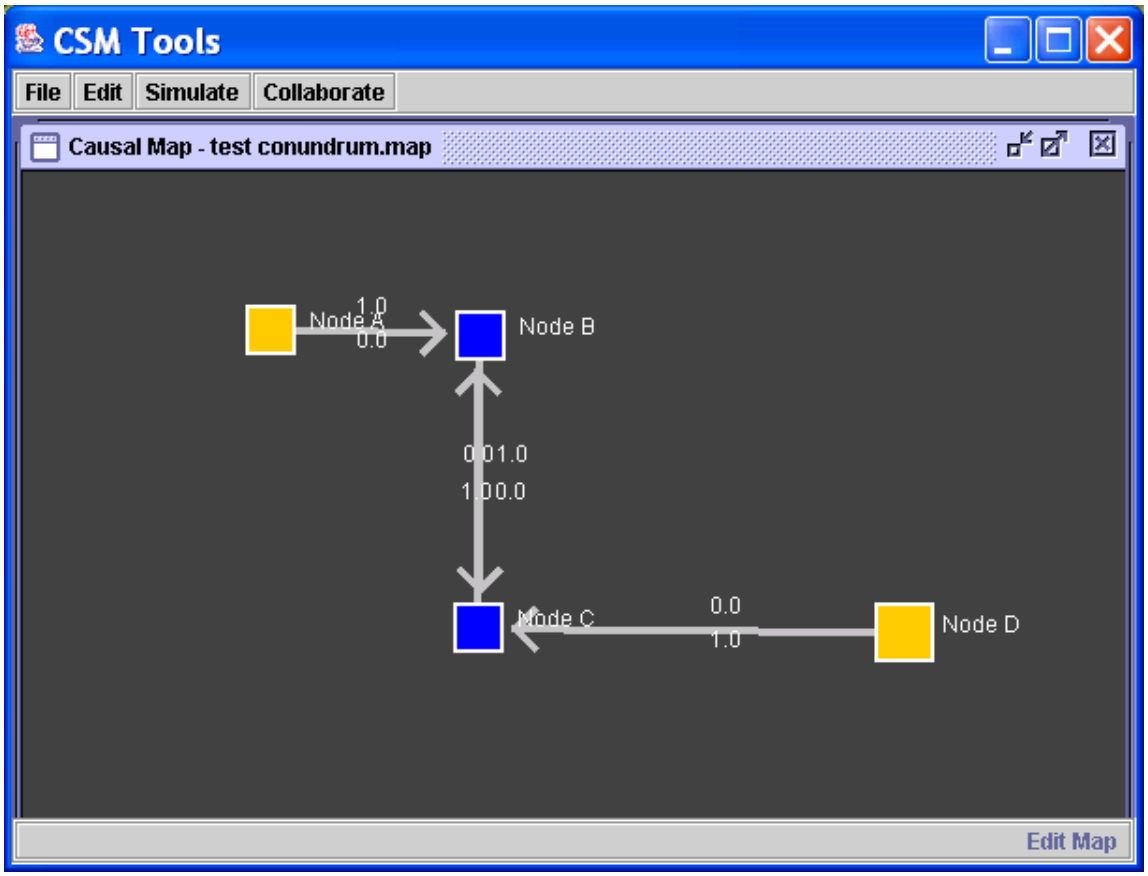


Figure 22. Test map of infinite causal loop.

Agents (nodes) are processed in order from A to D. In time step zero, agent A initiates a change to agent B. The associated channel is a full channel with a time lag of zero. The magnitude of the change coefficient is one. Since the time lag is zero the change is placed on the link to agent B and the link is immediately activated. Agent B is linked by a full channel with a time lag of zero and change coefficient of one to agent C. The link is immediately activated. Agent C has a return link to agent B and an incoming link from agent D. Agent D is a source of change and since it has not yet had a chance to act, agent C has received no change from D. The change from A via B is present and C processes the change and places a change on the return link to B. Agent D is then processed and a change is placed on the link to C and the link is activated. If agent behaviors are only scheduled for one time activation during a time step, all processing stops and the simulation time is advanced. However, processing is incomplete according to the semantics of the map. There still is an unprocessed change on the D-C link and on the C-B link. The nodes should be revisited to process these new changes. After processing, however, there

will now be a change present on the B-C link and a second change on the C-B link. This of course leads to an infinite loop and escalating change.

These feedback loops are common in causal maps and systems dynamics and, while problematic for automated processing, are essential to modeling complex systems. From the standpoint of scheduling, the infinite regress is what *should* happen given the semantics of the situation. Scheduling of agent behavior, therefore, needs to be recursive and not just a single pass polling of agents. Even without feedback loops, agent D's changes would not be processed by Agent C because of the sequential ordering. Random processing could be used, but with a single pass system the problem still remains. So recursive scheduling is necessary but can lead to the infinite causal loop problem.

A working solution to the problem can follow two basic approaches. One approach is to limit the semantics so that time lags of 0 are not allowed. This keeps the system from hanging and allows the loops to play out over time. Part of the problem is that we are using discrete units of time which could be one minute, one hour, one day, etc. Causal systems take place in continuous time but our representation is discrete. If the basic unit of time in the simulation is one month, a time lag of zero means something less than one month. True simultaneity is not a feature of causal networks. The cause is prior to the effect in social networks.

The second solution involves a more intelligent use of threshold values by the agent. Agents can keep track of time and only allow so much change during a single time step. This would allow a certain amount of recursion during a single time step but stop short of infinite progression.

CHAPTER 5 PROTOTYPE TESTING

Testing the prototype assesses its usability as a tool to draw and simulate causal maps. It is important at this point to delineate the surrounding process of the creation and formation of causal maps to frame strategic situations. The strategic quality of the maps or the effectiveness of causal mapping and CSM as a system for surfacing and testing of strategic assumptions is an area of further research. Much has already been accomplished in the work of Acar (1983). Thus the prototype testing performed in this research only assessed the completeness of the automated tool in correctly handling the semantics and processing of CSM and the ease of use for drawing and simulating a causal map once it is developed. It is beyond the scope of the current system to test how the map functions in the context of a group process and can be used as a facilitation tool in actual situations. This will require corresponding developments in a surrounding collaborative socio-technical system in which the causal mapping tool is embedded. Such a system was not part of this development and must be left to further research. Once the basic usability of the prototype is established, a system can be designed to embed the tool in a framework of processes that allows for further development of CSM as a complete group decision support system.

5.1 Semantics Testing

The purpose of the semantics testing was to construct maps that tested the basic semantics of the system. This involved insuring that the system accurately processed the intended semantics in terms of identification of triggers and levers, the type of channel, the correct transmission of change, and time synchronization. Most critical in this process was the testing of the type of channel and the correct transmission of change. Actual maps from Acar's (Acar, 1983) original case study were used as test maps.

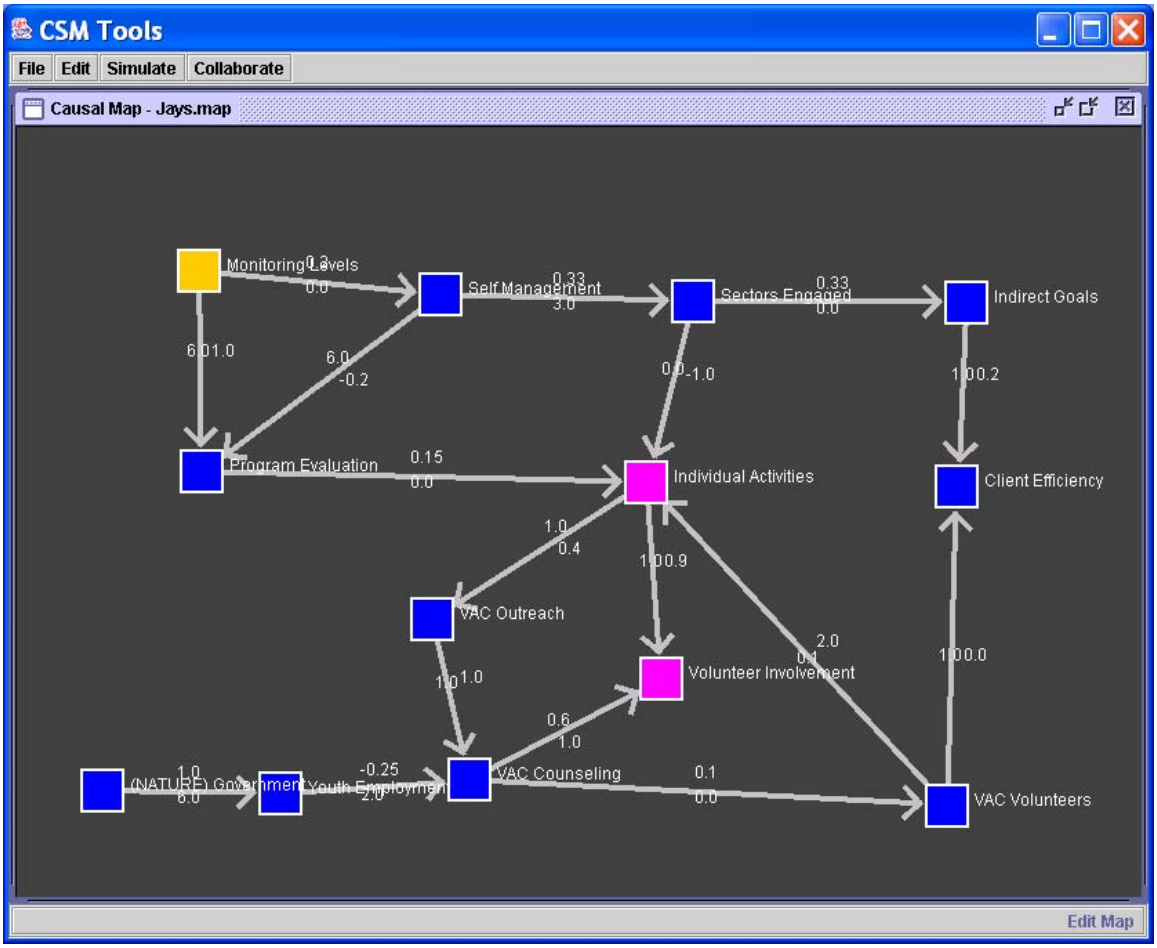


Figure 23. Jay's map

Jay's map (see Figure 23) was used initially as a test of the full channel semantic since it is composed entirely of full channels. This computerized map reproduces all of the semantics from the original case study and was used in the process testing phase to test the basic functionality of the simulation capability of the system.

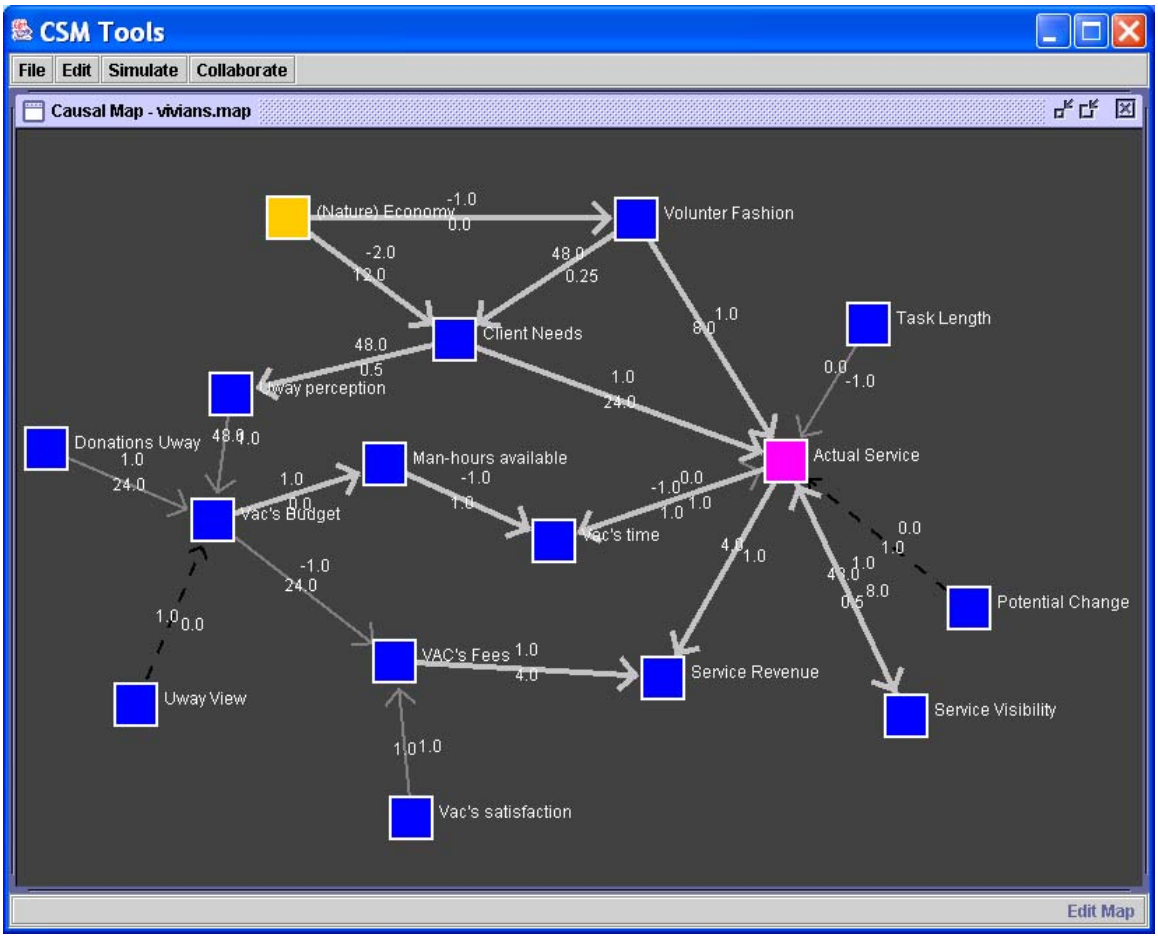


Figure 24. Vivian's map

In Acar's (Acar, 1983) original case study, Vivian's map was unique in that all link semantics were used in the map's construction. The map was recreated in the prototype system. The complete set of node and link semantics was captured in the map (See Figure 24).

5.2 Process Testing

Process testing tested the ability of the system to correctly create, store, retrieve, and simulate causal maps. This testing validated that the system could correctly simulate casual maps. Special test maps were also constructed to test extreme cases to assess the robustness of the system and to assess its ability to process any constructed map.

Jay's and Vivian's maps were also used as test maps to insure that the simulation correctly processes all types of channels. In both cases the basic goals of the individual were mapped and scenario A from the case study was simulated. The end result of each node after simulation was compared with the original case study to determine the correctness of the processing of full channel linkages. A comparison was also made at each moment in simulation time. In each case the results were the same as in the original study with the exception of Vivian's map. In the original calculations done by Acar (Acar, 1983), a small error was made in the manual calculations after several iterations of changes. This is not surprising given the tedious nature of manually calculating and tracking the changes through the scenario.

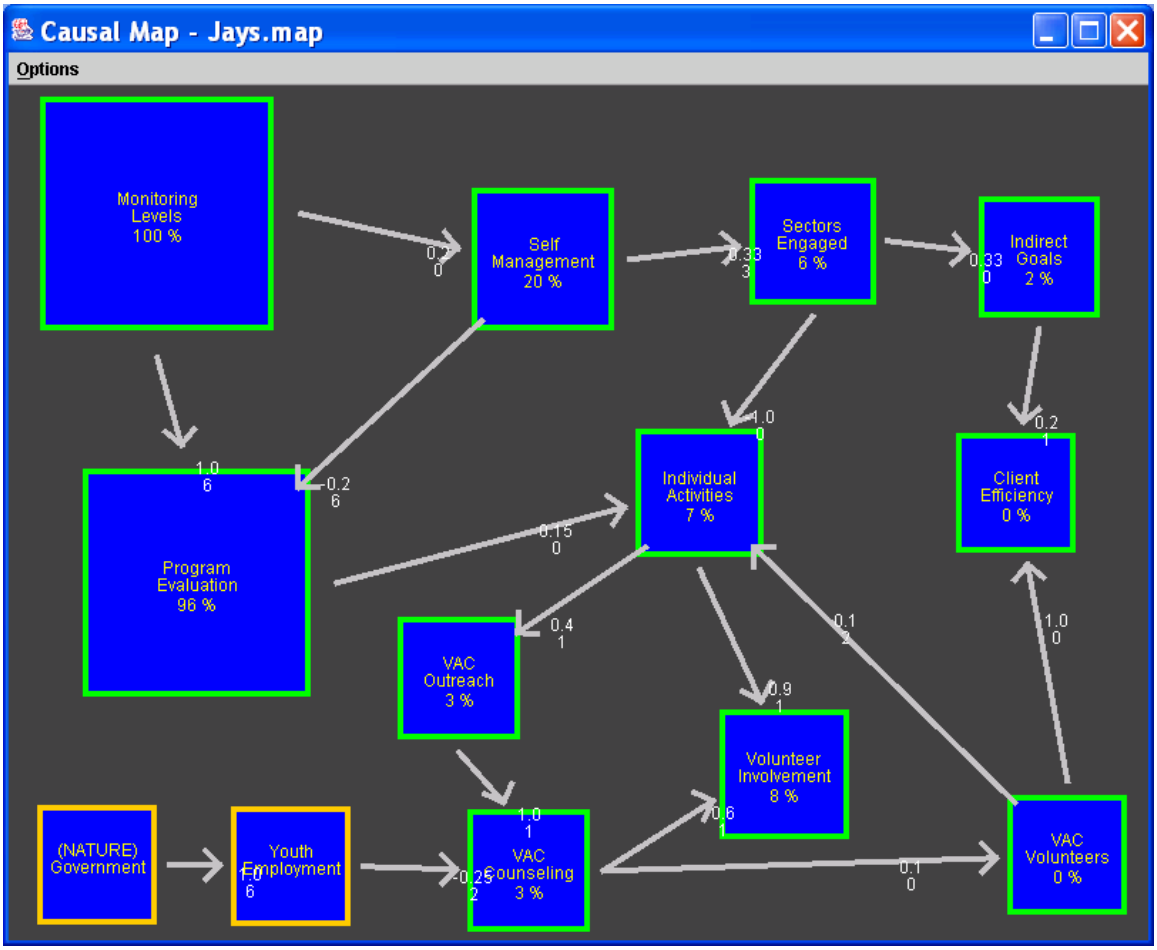


Figure 25. ***Scenario A*** **simulation results for Jay's map.**

Jay's map (see Figure 25) was also simulated with the assumptions of scenario A. The simulation stops at the end of 16 months of simulation time and all nodes have the same values as calculated in the original case study.

Figure 26 shows the goal nodes for Jay's map. At nine months most of the change has taken place in the system. After that point negligible change takes place and the system automatically stops after another seven months of simulation time. At this point the minimum default threshold stops the transmission of change through the map.

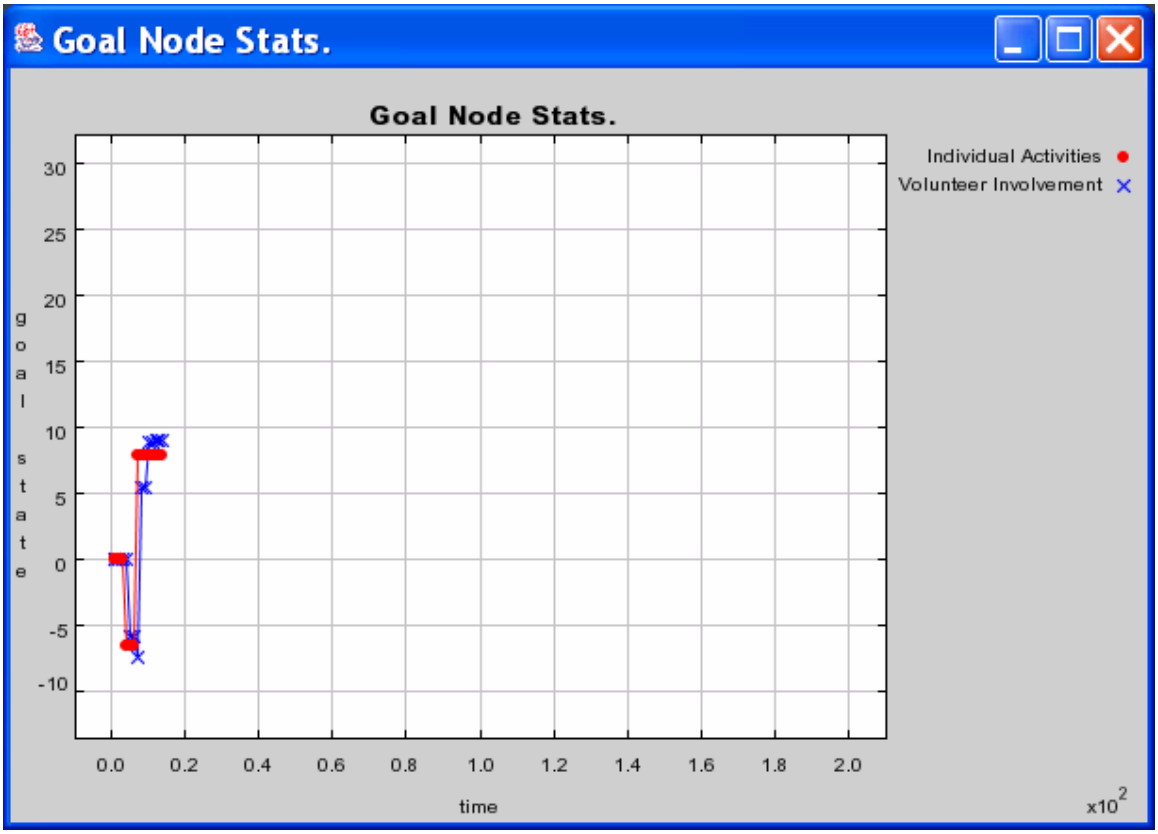


Figure 26. ***Scenario A*** **graphed goal nodes for Jay's map.**

Vivian's map was also used for testing proper processing. Of the maps in the original case study, it utilized all the available channel semantics. The simulation result of Vivian's map is shown in Figure 27.

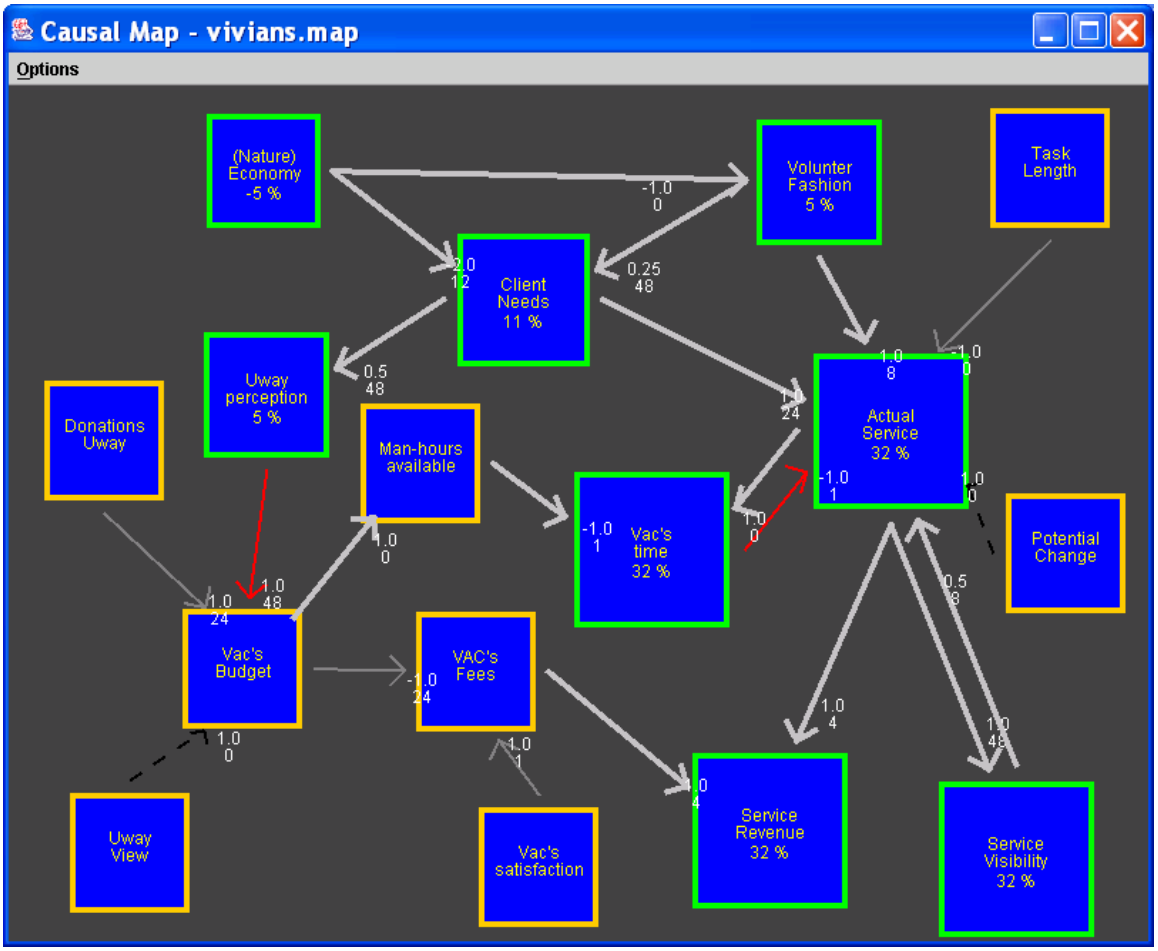


Figure 27. ***Scenario A*** **simulation results for Vivian's map.**

The map terminates after 120 months of simulated time (see Figure 28). The original case study was ended after 34 months of simulated time. The automated simulation of 120 months required 20 seconds to complete. Even the most exacting and meticulous hand calculations are prone to error and that is true in this case. In the original case study, a small calculation error, was made in the 32nd month. At this point the results of the computer simulation differ from the hand calculations. The tedious nature and complexity of the calculations, which are prone to error, was born out in the usability study addressed in the next section.

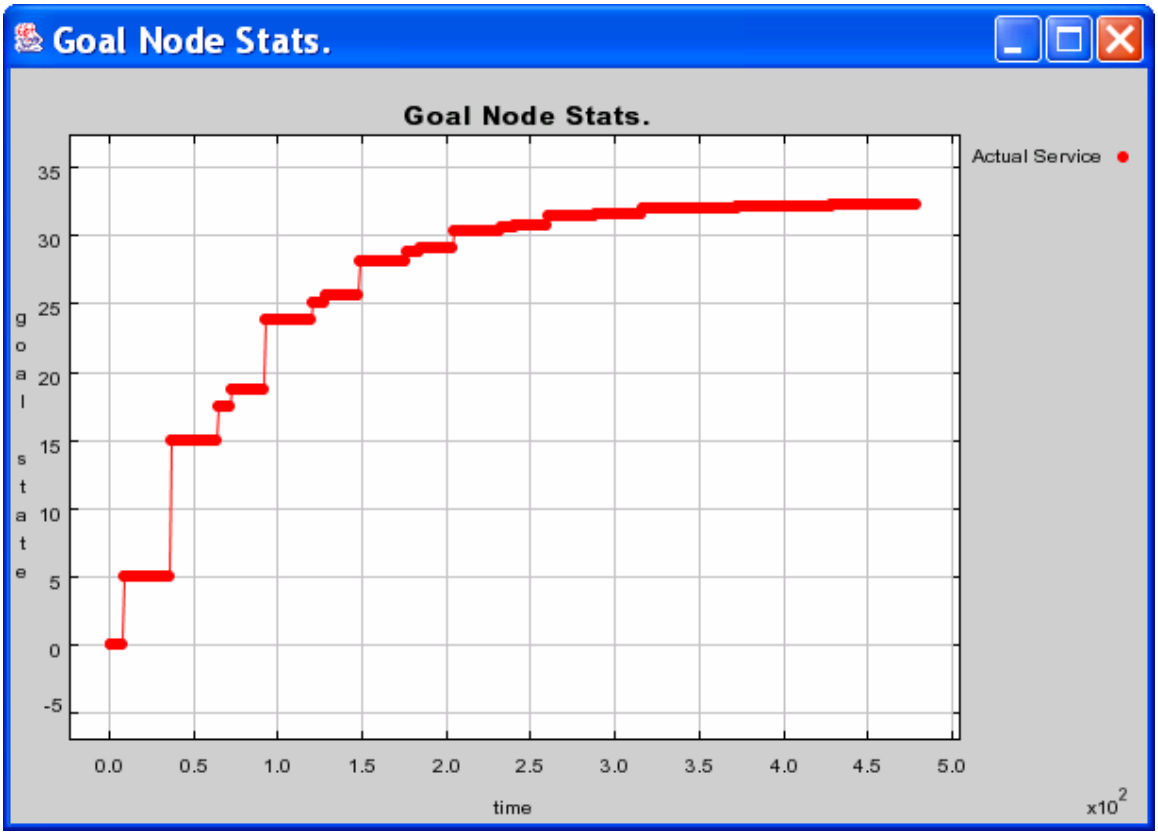


Figure 28. ***Scenario A*** **graphed goal nodes for Vivian's map. Time scale is in weeks.**

Partial and restriction semantics were also tested. For this test an abstract map was created to test proper simulation in isolation. Figure 29 shows the partial map. This map has three trigger nodes highlighted in yellow (nodes A-C and one goal node highlighted in magenta (node D.) There is also one restriction link and associated node highlighted in blue (node E.) The three partial links have time lags of 1, 2, and 3 respectively and change coefficients of 1, 2 and 3. The trigger nodes are also loaded with an initiating level above the status quo of 1%, 2%, 3% respectively. For the first test the restriction link was not triggered and the partial links worked properly. Subsequent testing tested the restriction node again with a proper result.

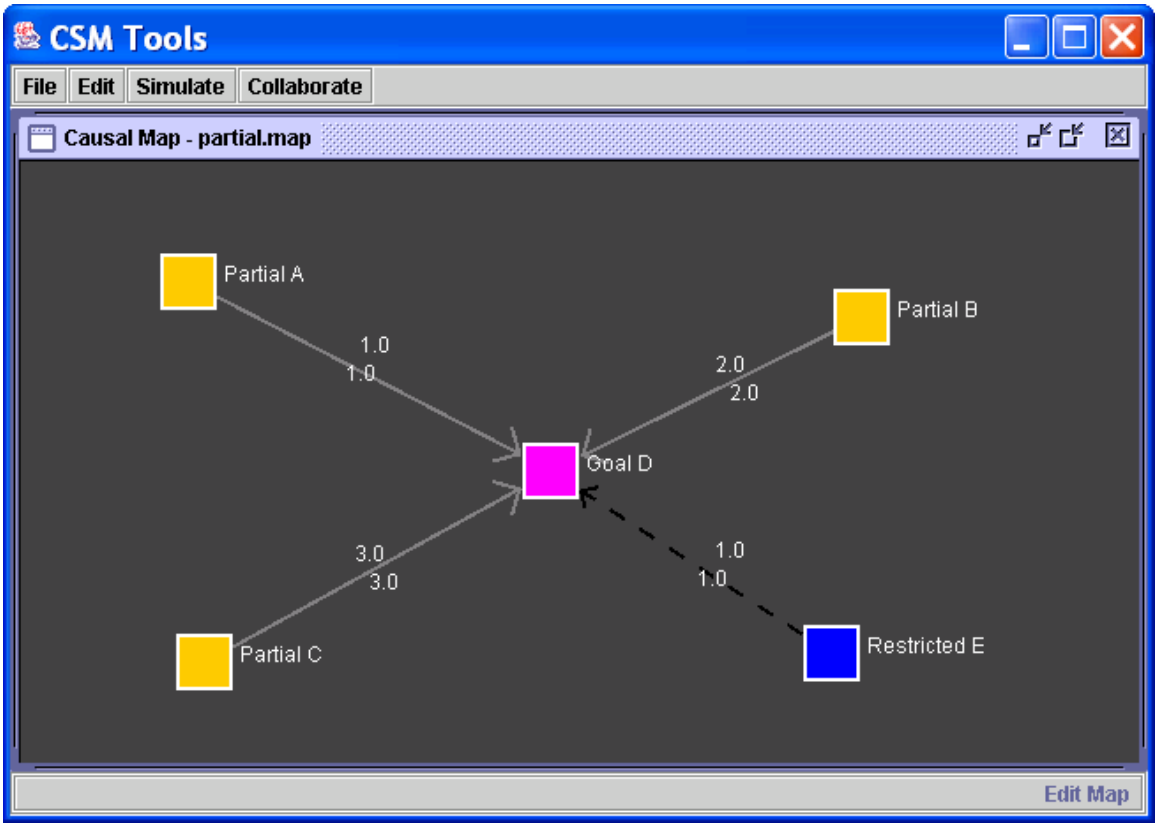


Figure 29. Test map for partial and restriction link types

5.3 Usability Testing

Usability testing provides feedback on the prototype that can be used to improve the system. The purpose of usability testing is to improve the system's ease of use, quality, user satisfaction, and user time on task. Common problems of usability testing include subject sampling (the generalizability of the results), test implementation (experimental design of the tests), validity and reliability (evaluation and coding of test tasks), and results interpretation (problems of experimental bias and subjective evaluation) (Zimmerman & Muraski, 1995). Formative evaluation aids the development of a prototype through a continual iteration of the test, revision, test cycle. Summative evaluations support revisions in future releases of already developed products (Porter & Coggin, 1995). Testing may take place either in the field or in a laboratory setting. State of the art usability test laboratories can cost as much as $100,000 and include two-way mirrors for subject observation, specialized video and audio equipment for

recording subject reactions, and specialized computer software for real-time data collection (Zimmerman & Muraski, 1995).

Data collection of evaluation results benefits from triangulation which uses a multi-method approach to minimize errors and maximize result benefits. Protocol analysis is most typically used. Typical tasks are defined and are given to subjects to complete. Subjects talk out loud while executing the tasks and the test is recorded onto video tape. The evaluators can then review the test, analyze the data, and interpret the results. Direct observation is also used, as well as personal interviews and surveys. Clearly identified variables and coding schemes can aid the evaluation and interpretation of data.

5.3.1 User Centered Design

Usability testing is part of an emphasis on User-Centered Design (UCD) (Rubin, 1994). UCD is a process that designs the relationship between software or hardware products and the user of those products with an emphasis on how the design enables user-centered tasks. This process is intimately connected to prototype development. The principles of UCD emphasize a development process that is iterative: early design focuses on users and tasks, empirical testing, and iterative design. Iterative design is a continual cycle of prototype design, modification, and testing. The emphasis on iterative UCD has shifted from a focus on technical implementation, which can now be automated or outsourced, to user interface design. The goal of UCD has been to fit the design of the system to the needs of the user, rather than fit the user to the needs of the system. The former support effective task accomplishment and are easy to learn, while the later are difficult to use and learn, require expensive training, and often result in low user acceptance.

Three components are integral to the development of usable systems:

- The user interface
- The help system
- The Documentation

The development of competitive products utilizes both an emergent approach of experiential learning, but this is accomplished in a rational framework. The goals of usability involve four factors:

- Usefulness. The first part of this study on strategic theory goes to the usefulness question. Causal mapping as a scenario creation process and the general theory of problem formulation address the issue of usefulness. Rubin points out that this

is the most overlooked element of UCD and leads to a machine-oriented, non-useable design.

- Effectiveness (ease of use). This emphasizes quantitative factors either by speed of performance or error rate and percentage of users
- Learnability. This refers to the ability of the user to operate the system at some defined level of competence after training.
- Attitude (likeability). The user's perceptions, feelings, and opinions of the product based on written or oral evaluation.

A number of methods for UCD are used, including participatory design, focus group research, surveys, design walk-throughs, paper and pencil evaluations, expert evaluations, usability audits, usability testing, field studies, and follow-up studies.

Usability studies are either formal summative studies conducted as true experiments with tested hypothesis or are formative assessments designed to expose usability deficiencies. The formative assessments are of four types: exploratory, assessment, validation, comparison. Comparison studies are most often done at the exploratory stage but can be combined with each of the other three stages of testing.

1. Exploratory testing of the prototype. Users are asked to perform standard tasks and to both describe how they would go about them and to actually try to accomplish them. This type of testing provides guidance on the design of high-level tasks. Only product mockups, paper designs or initial prototype with 30-40% functionality are necessary. Comparison testing of alternate prototypes is useful for high-level design innovation.

2. Assessment testing. This type of testing is a deeper level testing of user tasks that collects quantitative data on specific system functions. A single prototype is tested with 60-80% of required functionality.

3. Verification testing. Prior to final product release, verify that the system meets all functional requirements and make minor adjustments to the product before use by the end user. A full release product is necessary with 100% functionality, help system, and documentation.

5.3.2 Test Plan

5.3.2.1 Purpose

The purpose of the usability test is to ***explore*** the basic usability of the current prototype compared with the original paper based system. This testing is limited to exploration of user

interface issues in terms of its effectiveness, learnability and likeablity; establishing baseline data for the completion of common tasks; and generating suggestions for product improvement and further development of the prototype.

5.3.2.2 Objectives

- Effectiveness:
  - Discover problems the users encounter in creating and simulating causal maps, and the severity of those errors. Error level for each task is tracked by severity defined as:
    - The program hangs and must be aborted or is aborted unexpectedly.
    - Exceptions are reported in the command window associated with program execution, but the program does not hang or abort.
    - The user encounters a problem with task execution and requires more than 1 minute to solve the problem.
    - The user must ask for assistance in performing task.
  - Develop baseline statistics for the time to complete each of the basic tasks in both the original paper systems and the proposed prototype.
  - Measure the accuracy of computing basic scenarios both by hand and by the new system.
- Learnability:
  - Discover the minimal help and instructions needed to easily learn how to use the system.
    - Given a simple set of instructions, can users create basic map elements?
    - Can users repeat the creation of basic map elements with no additional instruction?
    - Does performance in use of the system improve over time?
  - Evaluate the modal design for switching between edit and add modes.
  - Discover the principle topics that will need to be included in the help system.
- Likeability:
  - Preference for using the manual system

- Creating maps
- Changing maps
- Scenario analysis

- Preference for using the computerized system
  - Creating maps
  - Adding elements
  - Editing elements
  - Saving/retrieving maps

5.3.2.3 User Profile

Users require no specific skills or abilities to use the software other than general familiarity with the common user interface requirements of windows applications. The intended user group of the prototype system is users with a general familiarity of windows products and user interface system; no special expertise or experience is required. Research suggests that 80% of usability problems can be detected with as few as four users. Additional subjects are less likely to discover more defects (Caulton, 2001). Discovering defects is only part of the study however and a larger sample size will be used to obtain baseline data and comparative statistics on the original manual system and its computerized counterpart.

The key determination of the number of users needed in the sample size is unique to usability testing and is based on the homogeneity of the subjects and the verticality of the application. If the subjects have discernable differences in experience with regard to the technology being used, they can be classified into subgroups. Due to the relative sophistication with the technology, each subgroup may uncover different kinds of problems. Larger sample sizes are then needed to uncover 80% of the usability problems that exist in the prototype (Caulton, 2001). The verticality of the software is also an issue. If the use of the software is narrowly defined for a specific use, its application is highly vertical and the homogeneity assumption is supported. Applications with horizontal or wide application by different types of users will need larger sample sizes to detect most problems. Since the tested prototype application has a narrow scope and user sophistication with Information Technology would have no affect on use, the assumption is that homogeneity in this case holds. A total of 16 participants from a senior level course for management information systems majors performed the testing during the weeks of June 21- July 2, 2005. The testers were relatively experienced with standard office software and a graphic user interface (GUI).

Sixteen test users were involved in the usability test. The testers were recruited from a special topics class in the summer of 2004. Testers received extra credit for their participation in the usability test. The university human subjects review board was contacted to insure compliance with research involving human subjects. Since the research was testing software, not human subjects, no special permission or review was required. In addition all testers were notified that they were not being tested; this was a test of the software to gain feedback for development and their participation was voluntary (See Appendix A.1 Introduction).

A background survey was conducted to characterize the background and experience of the users with a graphic user interface, graphic drawing tools and simulation software. In addition other descriptive and demographic data was collected indicating academic background, age, and sex. All testers were college seniors with 93.75% of them being Computer Information Systems majors. Seventy five percent (75%) of the testers were male and 56.25% were traditional aged students (age 21 to 23). Most testers (93.75%) had completed 4 or more information systems courses and had a high or very high comfort level (87.5%) in using a PC. All testers had more than four years experience using a graphic user interface such as Microsoft Windows. However, few testers (25%) classified themselves as regular or expert users of Microsoft powerpoint drawing tools and even fewer considered themselves expert in using other popular graphics editing or drawing tools software. Most testers (75%+) were infrequent users of graphics drawing tools similar to the test prototype. Only one tester had regular experience with software simulation or modeling tools.

The background of the testers is well matched to the usability study at this stage of the development of the prototype. This testing combined elements of both stage one (exploratory testing) and stage two (assessment testing) testing approaches. Test users were sophisticated enough to provide useful information on the interface design and recommend alternatives, uncover major flaws in the implemented prototype and provide an indication on the software's ease of use, especially in comparison to the original manual system.

A concern with the tested users is the generalizability of the test results to the intended target audience. The intended target for the eventual use of the software is top level managers and executives and the test users are relatively sophisticated young people with a high degree of comfort in the use of a personal computer. Much has been written about the technophobia of top executives (Connelly, 1994; Washbish, 2003). Anecdotal evidence and a search of the literature on executive computer phobia would indicate that this has changed substantially in the past few

years due to the pervasiveness of the Internet and the perceived strategic importance of IT for competitive advantage. Ralph Szygenda, CIO of General Motors, comments on the change in top executives' approach to becoming computer literate (Automotive News, 2000).

> "General Motors' top executives have gone from computerphobes who couldn't handle e-mail to Internet pros," says Ralph Szygenda, GM's chief information officer. "I'll take the senior management team of GM and we'll put them in a competition on laptops and Palm Pilots with any management team in the country right now," he said. "And I hate to tell you where we were two to three years ago. It was not good."

A now dated article (Martin, 2000) in Chief Executive "CEO's in Danger" lists technophobia as one of the seven deadly sins for CEO's in danger of loosing their job through poor performance.

> Being the CEO of a lumber company or an auto parts distributor isn't exactly glamorous these days. But it's downright dangerous if that CEO doesn't have a measure of computer literacy and an Internet strategy.

Since 2000, few articles have been written on the subject. The ubiquity of IT and its role in supporting a firm's strategy has brought an end to the illiteracy of the CEO. A review of current articles in ***The Chief Executive*** magazine indicates the technological sophistication of current CEO's. It is a reasonable assumption that today's current executive or top manager is computer literate, with at least four years of experience in using a graphic user interface based on Microsoft office products. This impacts the design of software interface for executive users. The design should follow standard conventions in following the common user interface (CUI) guidelines set by the industry. The test users fit this profile very nicely.

5.3.2.4 Task analysis

The basic tasks that need to be accomplished to use the prototype include:

5.3.2.4.1 Creating causal maps

Creating a new map

Saving a current map

Opening an existing map

Closing a current map

Node editing

- Node add
- Node delete
- Node properties
- Moving nodes

Link editing

Link add

Link delete

Link properties

Moving links

5.3.2.4.2 Simulating causal maps

Identification of goal nodes

Identification of trigger and lever nodes

Starting a simulation

Stepping through a simulation one time step at a time

Pausing a simulation

Stopping a simulation

Inspecting nodes

Map edit / simulate cycle

Make map changes

Re-simulate the map

5.3.2.5 Measurement selection and plan

Two types of measurement were used: direct observation and note taking during the testing process, time on task recording using a stop watch, and a follow-up in-depth interview to evaluate user satisfaction. Testing took place on a private office computer in an academic office area. Testing was done sequentially and on the same computer in the same environment. A test scenario was constructed that provided a sample causal map and specific tasks for both creating and simulating the map. The ABC sales scenario from Acar's (1983) original empirical testing of the system was used. It contained both a map and a text description of a situation that was easily understood by the user. This maintained consistency with earlier empirical studies of the method, is typical of a user created map and contains all the semantics of the Acar causal mapping system.

5.3.2.6 Data collection, analysis and recommendations

Data was collected during the usability testing and in a post test interview session immediately following the testing. Basic qualitative data was collected on errors encountered with the system and in the interview on the subject's satisfaction with the use of the system. Quantitative data was collected based on timings of assigned tasks that provide a basis for determining the learn-ability of the system. Data analysis was conducted to identify the primary

problems identified by test subjects and recommendations were made for proposed solutions and further development.

5.3.3 Test results

5.3.3.1 Effectiveness.

One of the major goals of the testing was to assess the effectiveness of the prototype in terms of uncovering hidden problems (bugs) in the current implementation of the software. The literature on this point was confirmed in that the majority of the problems with the software (80-90%) were uncovered by the first four users. After that point very few new problems were uncovered. In reality the testing was divided into two parts. A pilot group of users uncovered the major problems with the software, including hanging the program, and reported exceptions due to un-trapped errors in the program. After the pilot group, the test procedures had to be redesigned to work around several serious flaws in the program. This allowed other aspects of the prototype design to be tested without interrupting the testing process. In this regard the testing accomplished its objectives by exposing fatal flaws in the implementation leading to a much improved and more robust system.

Initially the test plan called for creation of a map, a scenario simulation, a redesign of the map and a second scenario simulation. This cycle represents the intended use of the product. The redesign of the map caused the second iteration of the simulation to fail. The map linkages in the simulation were scrambled and the second simulation was impossible to perform. This was problem was fixed after the testing was finished. A second major error was uncovered in simulating the test map “ABC Sales.” The partial channels in the map were being incorrectly evaluated leading to an error in the simulation results. This indicated a serious flaw in the processing of simulations and was also fixed after the testing.

Less serious problems with the interface, indicated by exception errors, were identified. These mostly revolve around the editing of property sheet input fields where received input is incorrect for the field type. These problems remain and need to be corrected before release and use of the software in field research.

The remaining problems uncovered by users have to do with the modal design of the interface. These problems are due to a design decision on how the map creation and editing functions should be performed and much of the prototype evaluation focused on this problem. Historically in user interface design of computerized editing tools, such as word processors or

drawing tools, two different approaches have been used in design. The older approach was based on a modal concept. This was necessitated by the use of the keyboard as the only interface to the program. The keyboard had to be used for both adding text to a document and submitting commands which controlled the editing features of the software (such as moving text.) Thus the user was required to shift between an "add" mode and an "edit" mode.

The current prototype uses this concept in its design. When nodes are added to the drawing surface, "add nodes" is selected. Each time the mouse button is pressed a node is added to the drawing surface. No editing (deleting nodes, calling up a property sheet, moving nodes around) is allowed. Since each click of the mouse adds another node, the screen can soon be covered with extra nodes if the user is not cognizant of this feature. A deliberate selection of the edit menu must be made to edit the map. Now the mouse can be used to select nodes, move them around, delete them, or edit their properties. Adding links follows the same pattern. This can be initially confusing since the alternative procedure is the one mostly used in today's graphic drawing programs. In this approach, once a node is added, the program automatically returns to edit mode. This means that to add another node, the "add node" item must be selected again. This involves extra key strokes and more time but represents the normal user interface employed in adding and editing objects in a graphic environment.

This interface was tested because it is an efficient way to construct causal maps composed of many nodes and links. Initially this created confusion with the users because of the difference with accepted practice. Users were provided documentation on how to use the application and were required to read the documentation before beginning the task of creating a map. A few users learned the interface with ease, some were initially confused and referred to the documentation to resolve the problem, most used trial and error to discover how the interface worked. Once the interface was learned, most users expressed satisfaction with the efficiency of drawing maps using this approach. Only one in sixteen users (6.25%) felt that "this was not the way to do things."

Baseline statistics for the time to complete each of the basic tasks in both the original paper system and the proposed prototype were collected. These allowed a comparison to be made between the manual method of drawing maps and construction of scenarios and the computerized approach. Automation of manual systems provides three benefits to the user: increased speed, accuracy, and consistency (Jessup & Valacich, 2003). Speed was measured by the time on task for creating maps, editing maps, and running a scenario. Accuracy was determined by the

correctly calculated ending value of a selected goal variable for a given scenario. Consistency is measured by both the appearance of the drawn maps and the consistent use of notation and its placement, resulting in a sharable and readable product.

5.3.3.1.1 Speed

A cursory look at the timings would suggest this benefit is suspect. The average time for users to draw a map by hand was 3:07 minutes compared with 10:04 minutes to create a map using the automated tool – 3 times slower! However this includes the time used to learn the interface without prior instruction or help (see Table 2).

Description	Average time in minutes:seconds	Standard Dev. In minutes:seconds
1) TASKS – Manual (n=15)		
a) Given the preceding description and the pencil and paper provided, draw the map including all descriptions and notations and Title	3:07	0:31
b) Develop scenario A	4:04	2:15
c) Develop scenario B	2:31	1:55
d) Modify the map	1:42	0:57
2) TASKS – Computer (n=15)		
a) Given the preceding description and the CSMTOOLS application, draw the map including all descriptions and notations and save as ABC SALES	10:04	3:05
b) Develop scenario A	2:22	1:18
c) Develop scenario B	0:47	0:15
d) Modify the map.	2:12	1:14
3) TASKS – Additional (n=15)		
a) Create a new blank map, redraw and save the sales map	4:38	1:17

Table 2. Table of average timings in minutes and seconds.

Once users learned the interface through creation of the map this step was repeated. The average time to draw a map the second time around was 4:38 minutes, comparable to drawing a map by hand but still the automated method is not an improvement in speed over drawing maps by hand. But the overall process must be considered. Maps are to be drawn, simulated, redrawn, and simulated again. The redrawing of maps and time to calculate scenarios must be considered in the timings. Calculating scenarios by hand involved learning effects as well. The average time to calculate a scenario the first time was 4:04 minutes. This compares with the first automated scenario calculation of 2:22 minutes, almost twice as fast. When a second scenario was constructed the manual approach still required 2:31 minutes while the automated scenario function required 47 seconds on average. The test scenarios were only a single pass through the map. Calculating scenarios further into the future and calculation of mixed scenarios with multiple triggers will further extend the advantage of the automated scenarios in terms of speed. When the whole cycle is taken into consideration, the speed advantage clearly goes to the automated system.

5.3.3.1.2 Accuracy.

This aspect was measured by an evaluation of the scenarios generated manually by the users. Users did have the aid of a calculator in making the calculations and assistance from the test administrator in understanding the process of calculating a scenario. The percentage of users who were able to calculate a scenario correctly by hand was 62.5%. Even experts in scenario calculation are bound to make some mistakes. In Acar's (1983) original case study there is a calculation error in Vivian's map. The likelihood of error in calculation is increased if the scenarios are mixed. This is no surprise and is the basis of most automation projects.

5.3.3.1.3 Consistency.

The consistency of the maps drawn by hand is irregular at best. This was measured by an evaluation of the hand-drawn maps with regard to the consistency of notation used in the scenarios and the readability of the resulting map. While 62.5% of the maps were neatly drawn, only 25% consistently applied the notation system in the construction of scenarios. This is another clear advantage of automating the system. A consistent notation system can be enforced by the program. The ability to easily edit the maps and change the arrangement of nodes and links is a clear advantage of an automated system.

5.3.3.2 Learnability.

The learnability objective was to 1) discover the minimal help and instructions needed to easily learn how to use the system, 2) evaluate the modal design for switching between edit and add modes and 3) discover the principle topics that will need to be included in the help system. This objective was evaluated through the observations made during the testing process. In several instances the users used the provided documentation to resolve problems with the interface, especially the modal design which required the user to switch between add and edit modes. The documentation provided the users could be considered minimal. While it was basically adequate for the creation of maps, additional documentation, possibly a tutorial on the creation of maps, would be useful in the opinion of several users. Many users prefer, however, to "discover" how the program works by using it rather than refer to documentation. Discovery learners can be assisted by further refinement of the interface to include visual clues as to the mode that is currently being used, such as a changed cursor shape or color. Quantitatively the learning curve can be measured by the average time it took to construct maps minus the time to construct a map once the system had been learned. On average the first time construction of a map using the system was 10:04 minutes (see Table 2). Recreating the same map later took on average 4:38 minutes. This means that on average 5:26 minutes were used in learning how to use the system to create a map.

Simulating scenarios is a more complicated problem, however. The written instructions for creating either manual or computer simulations were not enough for users to accomplish the task without assistance from the test administrator/helpdesk. First a short tutorial describing the processing of a scenario was given. Particular attention had to be given to the calculation of partial channels in manually calculating a scenario. Emphasis was also placed on proper notation and placement. Two scenarios were run, the first being a training run to give the user some directed experience in the process. A second run was made without assistance. The benefit of running a scenario by hand is an increased understanding by the user of the mechanics of scenarios. In the post-testing interviews, users all agreed that calculating scenarios by hand was beneficial in understanding the simulation aspects of the system.

Calculating scenarios using the automated system required a tutorial in how to set up the scenario with the software, how to use the scenario controls to affect the desired results and when to stop the scenario. Several problems with the software at this point became apparent. The scenario controls are preconfigured by Repast 2.0 and, while simple to use, require explanation.

Some less desirable aspects of the controls will require modifications to Repast. For example, multiple windows are created when a Repast simulation is run. The placement of the windows, many of which overlap others, is confusing to users. Closing one of the windows, instead of minimizing it, can abort the whole program. This results in a loss of the simulation, the map and any work not saved by the user. Another problem is the time-tick count in the simulator. The time tick always shows the next time-tick to be executed so that the current time is N-1. Users typically expect the time-tick to show the current time (actually the time period just completed).

The visualization system used in the prototype will need to be assessed in more detail. Exactly how best to present the scenarios visually is a prime area for further investigation. Currently the nodes change size to reflect the changes from the status quo. In non-terminating simulations the nodes can quickly get too large or too small, disrupting the display. Some type of scaling mechanism is needed that allows the display to be less chaotic. The goal node graph is an example of this. As node values become increasingly larger (or smaller) the graph is scaled to reflect this. The visualization of scenarios is both a clear advantage over the manual system and an opportunity for much additional research.

The time differential in creating and running scenarios was tracked to provide some baseline statistics on the learning curve and comparative analysis of the manual and automated versions of the system. Once the user completed the instruction on calculating a scenario by hand, the test scenario required 1:55 minutes to complete (see Table 2). This compares to 47 seconds for the computerized scenarios, which makes the computerized scenarios more than twice as fast. This advantage is dramatically increased in mixed scenarios (more than one input), successive iterations of scenarios (more than one iteration of the map), and in the visualization of scenarios that the software enables. All of these features contribute to the likeability of the software, discussed next.

5.3.3.3 Likeability:

In the post testing questionnaire and interviews, users exhibited a clear preference for the software over the manual process, especially with regard to the simulation of scenarios (see Table 3). The testing revealed that hand calculations of scenarios are tedious, time-consuming, and error-prone. When confronted with the task, one user exclaimed, “Oh great, math” and “Wow, I see why you made the software.” Another user in the midst of calculating a scenario decried “I’m completely lost,” and yet another commented, “This sucks! Now I know why we invented computers.” Users were asked in the questionnaire to rate the ease of use on a seven point Likert

scale. How easy was it to compute scenarios? The manual system received an average 3.88 on the scale which indicated that it was neither easy nor hard to manually compute scenarios. The automated system received an average 1.44 on the scale, which indicated that it was very easy or easy to compute scenarios with the computer. Overall the users liked the computerized system and either indicated that they preferred it over the manual system or would most likely use it as a tool.

n=16	Overall how easy was it to use paper and pencil?	Overall how easy was it to use the computerized product?	How easy was it to compute scenarios manually?	How easy was it to compute computerized scenarios?	How easy was it to create and edit maps using paper and pencil?	How easy was it to create and edit maps using the computer?
average	3.06	2.00	3.88	1.44	2.88	2.13
st. dev	1.12	0.73	0.96	0.63	1.20	1.15
min	2.00	1.00	3.00	1.00	1.00	1.00
max	5.00	3.00	6.00	3.00	5.00	5.00
p (t test)		.0018		.0000		.1038

Table 3. Seven point scale: comparison between manual and computer systems use.

Users were also asked to rate on a five point scale the likelihood of using the system as a tool for strategic planning. A comparison of the means from 11 respondents indicates that users were more likely to use causal mapping as a tool for strategic planning if it were computerized (see Table 4).

n=11	What is the likelihood you would use the manual system as a tool for strategic planning.	What is the likelihood you would use the computer system as a tool for strategic planning.
average	2.82	4.00
st. dev	1.13	1.17
Min	1.00	2.00
Max	5.00	5.00
p (t test)		.0836

Table 4. Five point scale: use of tool for strategic planning.

Of specific interest was the user's preference for the modal interface and the usefulness of the documentation. In post-testing interviews, users indicated a clear preference for the interface once it was learned. Only one user didn't like the interface. In general users found the

documentation useful but indicated a preference for a more complete description of the modal interface and how it worked. Despite the preference for running scenarios with the computer, all users felt that learning to calculate scenarios by hand contributed to their understanding of the system and its processes.

5.3.4 Recommendations

The usability revealed the shortcomings of the prototype's operation, documentation, and interface. These represent opportunities for further development and testing of the system. Several serious operational bugs were uncovered and fixed as a result of the testing. The first of these fixed a problem with the edit /simulate cycle where changes in maps were incorrectly processed by the simulation engine resulting in incorrect linkages between nodes in the simulated maps. The second major bug fix was an incorrect calculation of partial channels in the simulated test map.

Documentation and training will need further development. A detailed reference to the property sheet of nodes and links should be created that gives the users a complete description and tutorial on their meaning and use. A simulation tutorial needs to be created that includes detailed description and guidance on how to create and process a scenario using the system. This should include a description of how scenarios are calculated to aid the user's understanding of the system's output.

The interface requires substantial refinement. Data types for the property fields, field labels, input edit routines and error-trapping all require additional work. Editable global properties such as the time unit used, the size of the nodes, and color scheme, should be implemented. The placement of node labels within the node and the ability to enter them directly in the nodes without calling the property sheet would increase the clarity of the maps. Link notations need to allow the user to clearly distinguish between a time lag and a change coefficient. This can be accomplished through the use of different colors, time unit labels on time lags, and displaying only those link notations that differ from the default. The add/edit modal interface should be further developed and enhanced with a tool bar, improved graphics and visual indicators. Right clicking for a property sheet on map objects needs to be smooth and reliable.

These changes will need to be made before field testing and development can take place. Usability testing is a process of continuous improvement and once interface improvements are made they should be further tested and assessed.

CHAPTER 6 CONCLUSION

The research describes a practical linkage between scenario driven planning for strategic management and the latest distributed artificial intelligence modeling tools. Scenario driven planning is revolutionizing the way decision makers manage uncertainty and change. The development of an easy to use distributed tool for creating and simulating causal maps demonstrates the synergy of well designed methods for strategic thinking combined with intelligent support from DAI. It sets the stage for the development of a system using human/agent-based conceptual models to guide intelligent searches of internet, intranet and extranet space, thus combining ABM systems with more traditional MAS approaches.

6.1 Research Contribution

The research discussed in this paper underlies the feasibility of using agent-based modeling and simulation to support scenario development and simulation. It provides the basic tools for combining a user-friendly interface for graphically developing scenarios and simulation models. This puts the design of the simulation model squarely in the hands of the user of the system and provides for the collaborative development of scenarios via standard internet technologies. Until this development, computerized modeling and simulation was deemed too difficult for the ordinary user and was left to specialists who understood the technology but not necessarily the business situation. Decision makers themselves can now develop and exploit the power of agent-based modeling and simulation at minimum cost and maximum effectiveness. The research also opens the door to using an agent-based conceptual model to guide intelligent searches of internet, intranet, and extranet space to support backward analysis of causal factors, coefficients, and relationships. Given the wealth of information available from such sources, the development of a human-artificial conceptual map will be an invaluable guide to selecting relevant information for strategic decision making. This benefits both researchers and practitioners in several ways:

- Researchers in management and decision sciences benefit from the development of a practical research tool that integrates strategic theory with a distributed

artificial intelligence system permitting further investigation of decision makers' cognitive models and strategic decision formulation.

- Researchers in information systems will benefit from the demonstration of new integration technologies and application of distributed artificial intelligence solutions to complex and unstructured problem solving.
- Researchers in computer science will benefit from the development of an experimental system in the coordination of multi agent systems and human interactions.
- Management practitioners benefit from the research through development of a powerful support tool for strategic decision making that integrates knowledge management systems for strategic scenario development and foresight analysis.
- Researchers and practitioners in knowledge management systems benefit from the development of MAS knowledge representation techniques used in other socio-technical systems development and knowledge engineering projects.

6.2 Future Research

Many research questions remain. Key aspects of the system still need to be researched and developed.

- **Facilitation framework.** The larger context of Scenario-Driven planning must be carefully designed to both lead into and follow from the developed tool. The development of strategies from analyzed strategic scenarios and the refinement of the inquiry system that leads to the development of causal maps is an integral part of user acceptance. This indicates the need for a succession of field research projects using this methodology, refining the basic process and contributing to the development of an effective decision tool. Continued usability testing can assist in the development of the product and will likely entail increasing sophistication as the complexity of the system is developed. Additional features remain to be investigated and developed. How can different maps from different stakeholders be compared? What role can multi-agent systems play in making such comparisons and negotiating the process of consensus that allows divergent decision makers to converge mental models in a single causal map?

- **Management cognitive models and decision processes.** Causal maps are representations of a manager's perception of the underlying causal structure of strategic situations. Investigation of decision makers' cognitive models and strategic decision formulation in specific contexts contribute to the understanding of strategy and strategic theory. What is the relationship between managerial cognition and firm performance? What processes contribute to or counteract the phenomenon of group think that can lead to strategic blunders and disasters? To what degree does situational formulation solve this problem?
- **DAI integration and application**. The developed system is an experimental system in the coordination of multi agent systems and human interactions. Research is needed to extend the system capabilities, including intelligent searching of internet, extranet, and intranet space and the integration with standard BI and BAM technologies. How can the cognitive structures currently implemented as a set of linked agents be used to guide information searches that confirm or deny causal linkages and support key assumptions about change coefficients and time lags? How can cognitive models represented as agent structures be used to filter relevant information from the overwhelming amount of information in the environment?

APPENDIX A: USABILITY TEST DOCUMENTS

A.1. Introduction (read to all testers)

This test is a test of a prototype system that computerizes a manual scenario planning system. You will be given certain tasks to do – both manual and computerized. I will be observing you to see what problems develop with the computerized tasks and recording my observations. Tasks are timed, so I will tell you when to start and stop the task. Because we are comparing the computerized system with the manual one, you will be asked to repeat certain tasks. ***This is a test of the software and system, not a test of you***. If unexpected failures of the system occur, it is not your fault but is a good thing because it tells us what to fix in the application. Try to accomplish the task without help and refer to the documentation given you. If you need to call the help desk, talk to the observer of the test and ask for assistance.

A.2. Task Scenario – ABC Company Sales (Acar, 1983)

Please take 10 minutes to read the following text and study the diagrammed illustration on the next page.

For a number of reasons, a change in the sales level of company ABC would result in a corresponding change in net earnings equivalent to 60% of the sales change, in percentage.

A net earnings change generates, a year later, a five-fold change in ABC's credit limit. A change in credit limit will in turn induce, another year later, a proportional variation in ABC's advertising budget.

No one in ABC would say that the advertising budget can by itself influence the sales level. What is believed is that, together, variations in the advertising budget and in the perceived market share of the company would cause proportional variations in sales. However, this market share of ABC as perceived by the public will itself vary under the influence of the advertising budget; changes in perceived market share would be equivalent to $1/5^{th}$ of the advertising budget

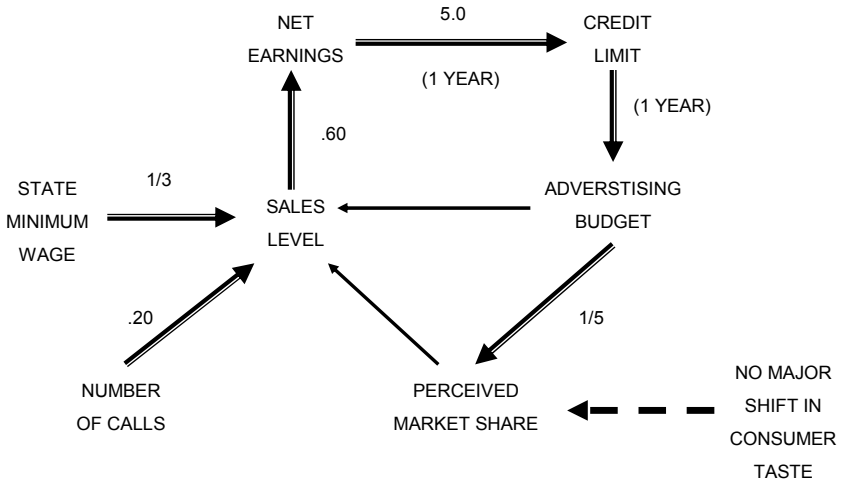
change in percentage. Whether this reduces the effectiveness of advertising expenditures is a debated question.

Changes in perceived market share do not automatically follow those in the advertising budget unless the following condition is satisfied; namely, that no major shift in consumer taste occur during the year.

In addition to the joint influence of advertising and market share, the sales level is subject to other influences. For one thing, a change in the state minimum wage would result in a three-times-smaller percentage change in sales. Also, a variation in the number of calls made by ABC's sales people induces a variation of the sales level equivalent to 20% of this variation.

This situation is graphically represented in a map on the next page. A short name has been written for each variable that the text describes as a possible source of change in itself or in the level of other variables. Factors which may prevent change under certain conditions (such as the "consumer taste") are also written down.

ABC COMPANY SALES



These factors and variables are linked by three types of arrow:

1. **Full Channel.** A double-line arrow is used to connect a sender and a receiver of change when the sender of the arrow is sufficient, by itself, to transmit a change to its receiver (e.g., each of the "state minimum wage" and "number of calls" being capable of inducing on its own a change in the "sales level").
2. **Half Channel.** Single-line arrows are used when two or more senders have to vary in order to generate a change in a receiver (such as the way in which the "advertising budget" and "market share" together determine the "sales level" of ABC).
3. **Restriction Channel.** When the sender is a factor which could prevent the receiver from changing under any influence (such as the condition "no major shift in consumer taste"), a dotted arrow is used to express this condition.

When the transmission of change among variables involves a time lag, or when the change induced in the receiver of an arrow is not comparable in sign or proportion to the change generating it, the extent of the time lag and the change equivalence coefficient are written next to the arrow (the equivalence coefficients express the ratio of the induced percentage change to the one generating it. If the change equivalence coefficient is not explicitly stated, it is assumed to be 1. If the time lag is not explicitly stated it is assumed to be 0.

APPENDIX B: TEST TASKS

B.1. TASKS - Manual System

B.1.1 Map Creation

- Given the preceding description and the pencil and paper provided, draw the map including all descriptions and notations.
- Title the map "ABC SALES"

B.1.2 Scenario Creation

Scenarios are created by selecting a trigger node (source of change), increasing or decreasing its default value by a desired amount and writing the change next to the node. Follow the outgoing arrows to any linked nodes. Multiply the sending nodes value by the link change coefficient to calculate the receiver nodes new value. Write down the receiver node's new value next to the node. If the link has a time lag, write down the time lag next to the node value. Repeat for each node. Use the windows calculator as an aid. Calculate all percentage values in whole numbers. Convert change coefficients to decimal numbers. (For example 1/3 is .33, 1/5 is .20, etc.)

- Develop scenario A by:
  - Increasing the value of the trigger node "STATE MINIMUM WAGE" by 20%.
  - Calculate the increase or decrease in "SALES LEVEL" at the end of 2 years.
  - Label calculated node values with A (for example A - 5.5 – 0, where 0 is the time step)
- Develop scenario B by:
  - Increasing the value of the trigger node "NUMBER OF CALLS" by 30%
  - Calculate the increase or decrease in "SALES LEVEL" at the end of 2 years.
  - Label calculated node values with B. (for example B - 5.5 – 0, where 0 is the time step)

B.1.3 Map Modification Modify the map by:

- Deleting the “credit limit” node and two associated links.
- Adding two new nodes “credit score” and “available credit”
- Adding three new full channel links:
  - Between “NET EARNINGS” and “CREDIT SCORE” with a change coefficient of 5.
  - Between “CREDIT SCORE” and “AVAILABLE CREDIT” with no additional notation.
  - Between “AVAILABLE CREDIT” and “ADVERTISING BUDGET” with no additional notation.

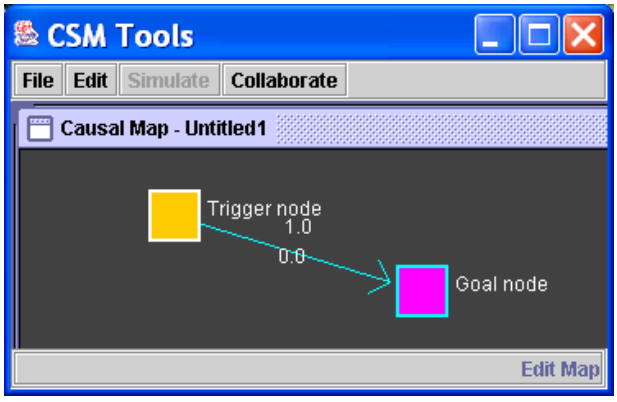
B.2. TASKS - Computer System

B.2.1 Map creation

- Given the preceding description and the CSMTOOLS application, draw the map including all descriptions and notations.
- Save the map with the name “ABC SALES”

B.2.2 Scenario Calculation

- Modify the property sheet of trigger node (source of change), change the default value to the desired level and commit the changes. The node color will change to yellow.
- Modify the property sheet of any goal nodes, click the check box “Goal Node” and commit the changes. The node color will change to magenta. (Goal nodes are charted when a simulation is run to allow the user to visualize the change of the node as the simulation progresses.)



- Select the Run Simulation item from the Simulate menu.

B.2.3 Senario Creation

- Develop scenario A by:
    - Increasing the value of the “STATE MINIMUM WAGE” by 100%
    - Calculate the increase or decrease in “SALES LEVEL” at the end of 2 years.
- Develop scenario B by:
    - Increasing the value of the trigger node “NUMBER OF CALLS” by 50%
    - Calculate the increase or decrease in “SALES LEVEL” at the end of 2 years.

B.2.4 Map Modification Modify the map by:

- Deleting the “credit limit” node and two associated links.
- Adding two new nodes “credit score” and “available credit”
- Adding three new full channel links:
    - Between “NET EARNINGS” and “CREDIT SCORE” with a change coefficient of 5.
    - Between “CREDIT SCORE” and “AVAILABLE CREDIT” with no additional notation.
    - Between “AVAILABLE CREDIT” and “ADVERTISING BUDGET” with no additional notation.

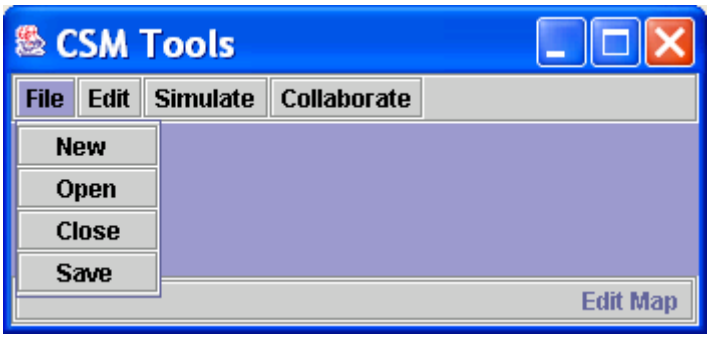
B.3. TASKS - Additional Computerized

- Create a new blank map
- Redraw the ABC SALES map
- Save the map as ABC SALES REDRAW

APPENDIX C: CSMTOOLS USER GUIDE

Take 5 minutes to study the diagrams below which explain how to use the computerized application for creating maps.

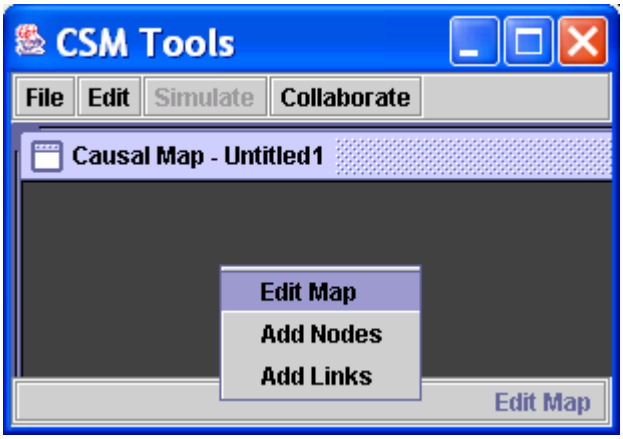
The File menu allow you to create “New” maps, “Open” existing maps, “Close” the current map, “Save” the current map.



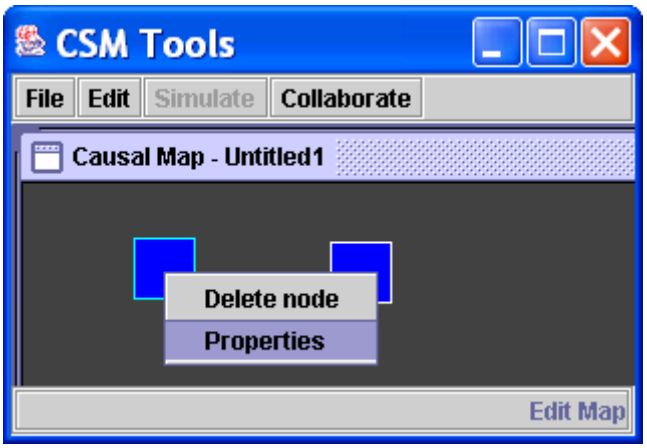
The “Edit” menu allows you to Edit the map, Add Nodes, and Add Links. In the “Edit map mode” you can select and reposition nodes and links on the map. When “Add Nodes” is selected, you can add nodes with each left mouse click. When “Add Links” is selected you can add links between nodes by left clicking the first node and then dragging the link to the receiving node.



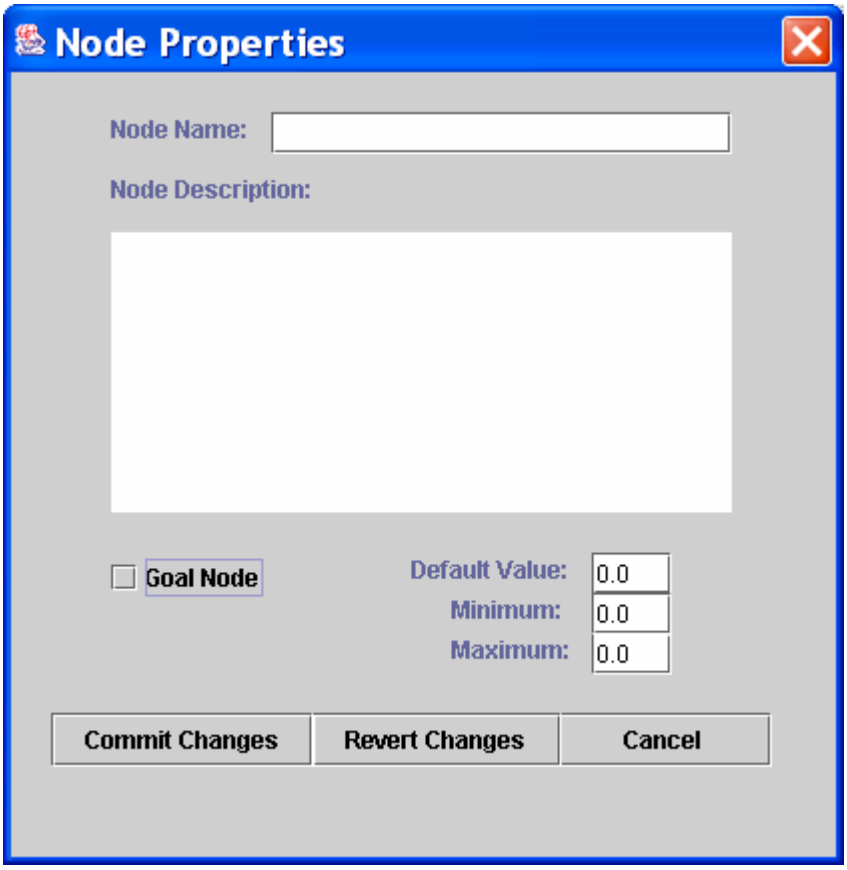
When a new map is created a map drawing surface is created with the name “Untitled1”. Right clicking on the grey drawing surface will present a pop-up menu with the same options as the “Edit” menu.



In “Edit Map” mode (indicated on the status bar on the lower right corner) right clicking on a node allows the user to edit the properties of a node or to delete the node.



Choosing the “Properties” option brings up the node property sheet.



Fields are available to enter a:

Node Name.

Node Description.

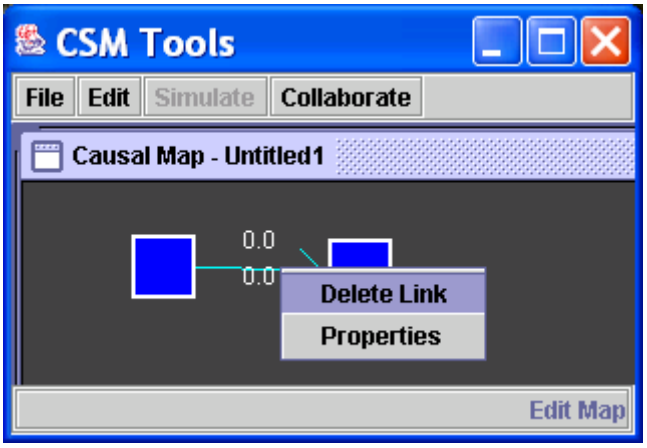
Default Value.

Minimum and Maximum threshold values.

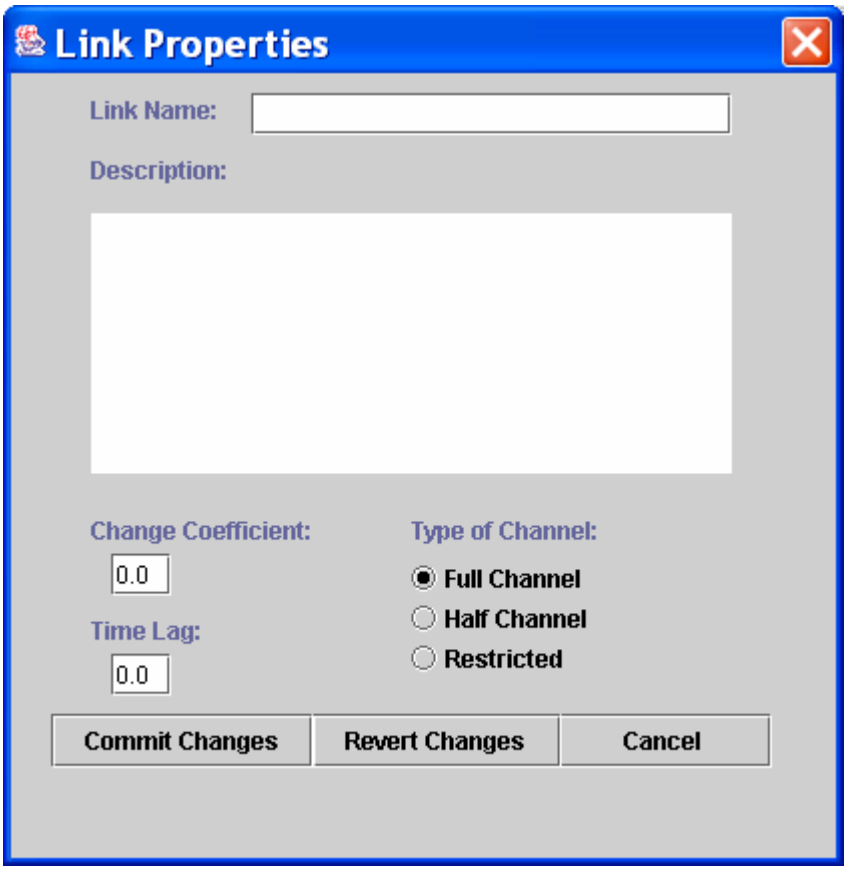
Goal node. (Used for simulation)

Buttons allow changes to be saved (Commit Changes), discarded (Revert Changes) or discarded and exit the property sheet (Cancel).

Right clicking a link allows the user to edit the link property sheet or delete the link.



The link property sheet pops up when the user chooses “Properties”



Fields are available to enter a:

Link Name.

Link Description.

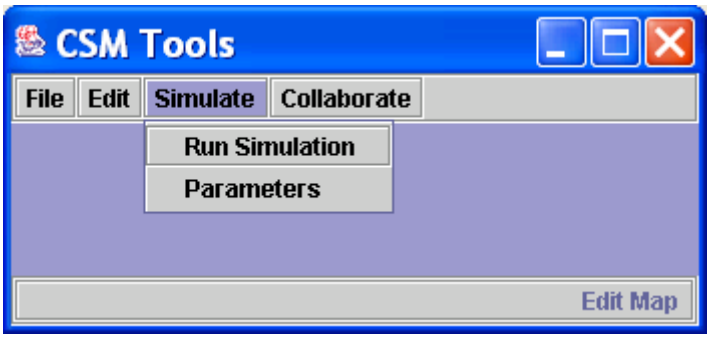
Change Coefficient.

Time Lag.

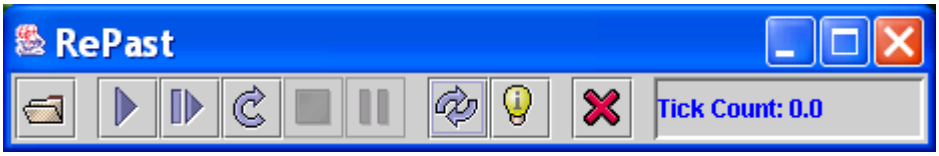
Type of Channel.

Buttons allow changes to be saved (Commit Changes), discarded (Revert Changes) or discarded and exit the property sheet (Cancel).

The “Simulate” menu allows you to simulate the map with the “Run Simulation” item. The “Parameters” menu item is not used.



When the “Run Simulation” item is chosen the simulation tool bar is displayed.



 Run the simulation until it stops.

 Step through the simulation 1 time step at a time.

 Stop the simulation.

 Pause the simulation.

 Restart the simulation.

 Exit the simulation function.



Next time step to be executed. (For example, if the next time step to be executed is 2 years, the tick count will show 2.0.)

Other icons  are unused.

APPENDIX D: POST TESTING SURVEY

Post-test questionnaire User: xx

1. Overall how easy was it to use the computerized product?
    1 - very easy
    2 - easy
    3 - somewhat easy
    4 - neither easy or hard
    5 - somewhat hard
    6 - hard
    7 - very hard

2. Overall how easy was it to use paper and pencil?
    1 - very easy
    2 - easy
    3 - somewhat easy
    4 - neither easy or hard
    5 - somewhat hard
    6 - hard
    7 - very hard

3. How easy was it to compute scenarios manually?
    1 - very easy
    2 - easy
    3 - somewhat easy
    4 - neither easy or hard
    5 - somewhat hard
    6 - hard

7 - very hard

4. How easy was it to compute scenarios using the computerized version?
    1 - very easy
    2 - easy
    3 - somewhat easy
    4 - neither easy or hard
    5 - somewhat hard
    6 - hard
    7 - very hard

5. How easy was it to create and edit maps using paper and pencil?
    1 - very easy
    2 - easy
    3 - somewhat easy
    4 - neither easy or hard
    5 - somewhat hard
    6 - hard
    7 - very hard

6. How easy was it to create and edit maps using the computer?
    1 - very easy
    2 - easy
    3 - somewhat easy
    4 - neither easy or hard
    5 - somewhat hard
    6 - hard
    7 - very hard

7. What is the likelihood you would use the manual system as a tool for strategic planning?

1 - Would definitely not use

2 - Would probably not use

3 – Not sure

4 - Would probably use

5 - Would definitely use

8. What is the likelihood you would use the computer system as a tool for strategic planning?

1 - Would definitely not use

2 - Would probably not use

3 – Not sure

4 - Would probably use

5 - Would definitely use

Comments

With the computerized version of the system what did you find:

Most confusing?

Easiest to do?

Hardest to do?

Usefulness of the instructions for the computer tool?

REFERENCES

Acar, M. (2000). "A Computerized, Collaborative System for the Creation and Edition of Causal Maps." B.S. Kent State University Honors College.

Acar, W. (1983). "Toward a Theory of Problem Formulation and the Planning of Change: Causal Mapping and Dialectical Debate in Situation Formulation." Ann Arbor, Michigan:U.M.I..

Acar, W. (1987). "A Biased View of Problem Formulation." In (pp. 116-118). Boston: Decision Sciences Institute National Conference.

Ackoff, R. L. (1967). "Management Misinformation Systems." **Management Science,** 11.

Ackoff, R. L. (1978). ***The Art of Problem Solving***. New York: Wiley.

Ackoff, R. L. (1981). ***Creating the Corporate Future***. New York: Wiley.

Amit, R. & Schoemaker, P. J. H. (1993). "Strategic Assets and Organizational Rent." **Strategic Management Journal,** 14**,** 33-46.

Ansoff, H. I. (1965). ***Corporate Strategy***. New York: McGraw-Hill.

Ansoff, H. I. (1991). "Critique of Henry Mintzberg's 'The Design School: Reconsidering the Basic Premises of Strategic Management'." **Strategic Management Journal,** 12**,** 449-461.

Automotive News (2000, October 23). "Szygenda Throws Down the Gauntlet." **Automotive News, 75,** 2.

Barney, J. (1991). "Firm Resources and Sustained Competitive Advantage." **Journal of Management,** 17**,** 99-120.

Barney, J. (2001). "Is the Resource-Based "View" a Useful Perspective for Strategic Management Research? Yes." **The Academy of Management Review,** 26**,** 41-56.

Boulding, K. E. (1969). ***The Image: Knowledge in Life and Society***. (7th printing ed.) Ann Arbor: The University of Michigan.

Bresnahan, T. F., Brynjolfsson, E. , & Hitt, L. M. (2002). "Information Technology, Workplace Organization, and the Demand for Skilled Labor: Firm-level Evidence." **Quarterly Journal of Economics, 117,** 339-376.

Caulton, D. A. (2001). "Relaxing the homogeneity assumption in usability testing." **Behavior and Information Technology,** 20**,** 1-7.

Chaib-draa, B. (2002). "Causal Maps: Theory, Implementation, and Practical Applications in Multiagent Environments." **IEEE Transactions on Knowledge and Data Engineering,** 14**,** 1201-1217.

Chakravarthy, B., Mueller-Stewens, G., Lorange, P., & Lechner Christoph (2003). "Shaping, Implementing, and Changing Strategy: Opportunities and Challenges." In B.Chakravarthy, G. Mueller-Stewens, P. Lorange, & Lechner Christoph (Eds.), ***Strategy Process*** (Malden, MA: Blackwell Publishing Ltd.

Chandler, A. D. (1962). ***Strategy and Structure***. Cambridge, MA: MIT Press.

Churchman, C. W. (1979). ***The Systems Approach and its Enemies***. New York: Basic Books.

Churchman, C. W. (1971). ***The Design of Inquiring Systems: Basic Concepts of Systems and Organization***. New York: Basic Books.

Connelly, J. (1994, November 14). "Why I fear and loathe my computer." **Fortune, 130,** 263-265.

Daniels, M. (1999). "Integrating Simulation Technologies With Swarm." http://www.santafe.edu/~mgd/anl/anlchicago.html [On-line]. Available: http://www.santafe.edu/~mgd/anl/anlchicago.html

Druckenmiller, D. A. (1991). "Face-to-Face Group Decision Support Systems: A Partnership Between Traditional Group Process Methods and Computer Technology." **Journal of Information Systems Education,** 3**,** 27-31.

Eden, C. & Ackermann, F. (1998). ***Making Strategy***. London: Sage.

Eden, C. & Ackermann, F. (2004). "Cognitive mapping expert views for policy analysis in the public sector." **European Journal of Operational Research,** 152**,** 615-630.

Eisenhardt, K. M. & Martin, J. A. (2000). "Dynamic Capabilities: What are They?" **Strategic Management Journal,** 21**,** 1105-1121.

Fernandes, R. & Simon, H. A. (1999). "A study of how individuals solve complex and ill-structured problems." **Policy Sciences,** 32**,** 225-245.

Forgionne & Kohli (2001). "Management Support System Effectiveness: Further Empirical Evidence." **Journal of the Association of Information Systems**.

Forrester, J. W. (1961). ***Industrial Dynamics***. Cambridge, Mass: M.I.T. Press.

Gassman, B. (2003). "Business Activity Monitoring Client Issues for 2004." (Rep. No. **K-20-9641**). Gartner Research.

Georgantzas, N. C. & Acar, W. (1995). ***Scenario-Driven Planning: Learning to Manage Strategic Uncertainty***. (1st ed.) Westport Connecticut: Quorum Books.

Godfrey, P. C. & Hill, C. W. L. (1995). "The Problem of Unobservables in Strategic Management Research." **Strategic Management Journal,** 16**,** 519-533.

Gorry, Anthony, G., & Morton, M. S. (1971). "A Framework for Management Information Systems." **Sloan Management Review**.

Hall, R. (1992). "The Strategic Analysis of Intangible Resources." **Strategic Management Journal,** 13**,** 135-144.

Harary, N., Norman, R. Z., & Cartwright, D. (1965). ***Structural Models: An Introduction to the Theory of Directed Graphs***. New York: Wiley.

Heintz, T. J. & Acar, W. (1994). "Causal Modeling as a Tool for Problem Framing within a GDSS: An Object-Oriented Approach." **Information Systems Journal,** 4**,** 291-310.

Hofstadter, D. R. (1996). "Analogy-Making, Fluid Concepts, and Brain Mechanisms." In Clark & Millican (Eds.), ***Connectionism, Concepts, and Folk Psychology: The Legacy of Alan Turing Volume II*** (pp. 195-235). Oxford: Clarendon Press.

Holland, J. H. (1992). ***Adpatation in Natural and Artificial Systems***. MIT Press, Cambridge Massachusetts.

Hoopes, D. G., Madsen, T. L., & Walker, G. (2003). "Why is There a Resource-Based View? Toward a Theory of Competitive Heterogeneity." **Strategic Management Journal,** 24**,** 889-902.

Huff, A. S. (1990). ***Mapping Strategic Thought***. England: Wiley & Sons.

Jennings, N. R. (2000). "On agent-based software engineering." **Artificial Intelligence,** 117**,** 277-296.

Jennings, N. R., Sycara, K., & Wooldridge, M. (1998). "A Roadmap of Agent Research and Development." **Autonomous Agents and Multi-Agent Systems,** 1**,** 7-38.

Jessup, L. & Valacich, J. (2003). ***Information Systems Today***. Upper Saddle River, NJ: Prentice Hall.

Kelly, G. A. (1955). ***The Psychology of Personal Constructs***. New York: WW Norton.

Lockett, A. & Thompson, S. (2001). "The Resource-based view and economics." **Journal of Management,** 27**,** 723-754.

Makadok, R. (2001). "Toward a Synthesis of the Resource-Based and Dynamic-Capability Views of Rent Creation." **Strategic Management Journal,** 23**,** 387-401.

Martin, J. (2000, October 1). "CEO's in Danger." **The Chief Executive**.

Maruyama, M. (1963). "The second cybernetics: deviation-amplifying mutual causal processes." **American Scientist,** 51**,** 164-179 & 250-256.

Mason, R. O. & Mitroff, I. I. (1981). ***Challenging Strategic Planning Assumptions***. New York: John Wiley & Sons.

McCoy, D. & Govekar, M. (2002). "Evolving Interaction Styles in Business Activity Monitoring." (Rep. No. **COM-17-8576**). Gartner Research.

McEvily, S. K. & Chakravarthy, B. (2002). "The Persistence of Knowledge-Based Advantage: An Empirical Test for Product Performance and Technological Knowledge." **Strategic Management Journal,** 23**,** 285-305.

Mintzberg, H. (1990). "The Design School: Reconsidering the Basic Premises of Strategic Management." **Strategic Management Journal,** 11**,** 171-195.

Mintzberg, H. (1991). "Learning 1, Planning 0: Reply to Igor Ansoff." **Strategic Management Journal,** 12**,** 463-466.

Mintzberg, H. (1994). "The Fall and Rise of Strategic Planning." **Harvard Business Review,** January-February**,** 107-114.

Mintzberg, H., Lampel, J., Quinn, J. B., & Sumantra, G. (2003). ***The Strategy Process***. (Global Fourth Edition ed.) Upper Saddle River: Prentice Hall.

Newell, A. (1982). "The Knowledge level." **Artificial Intelligence,** 18**,** 87-127.

O'Brien, J. (2002). "E-Business Decision Support." In ***Management Information Systems: Managing Information Technology in the E-Business Enterprise*** (5th ed., McGraw-Hill Irwin.

O'Reilly III, C. A. & Pfeffer, J. (2000). "Cisco Systems: Acquiring and Retaining Talent in Hypercompetitive Markets." **Human Resource Planning** 38-52.

Panepento, P. (2000, August 14). "The Perfect Swarm." **Computerworld, 34**.

Peteraf, M. A. (1993). "The cornerstones of competitive advantage: A Resource-Based View." **Strategic Management Journal,** 14**,** 179-191.

Phelan, S. E. (1997). "Using Artificial Adaptive Agents to Explore Strategic Landscapes." **La Trobe University, Bundoora, Australia**.

Porter, L. R. & Coggin, W. (1995). ***Research Strategies in Technical Communication***. New York: John Wiley and Sons, Inc.

Porter, M. E. (1980). ***Competitive Strategy: Techniques for Analyzing Industries and Competitors***. New York: Macmillian Publishing Co., Inc.

Porter, M. E. (1985). ***Competitive Advantage: Creating and Sustaining Superior Performance***. New York: Macmillian Publishing Co., Inc.

Porter, M. E. (1996). "What is Strategy?" **Harvard Business Review,** November-December**,** 61-78.

Porter, M. E. (2001). "Strategy and the Internet." **Harvard Business Review,** March**,** 63-78.

Powell, T. C. (2001). "Competitive Advantage: Logical and Philosophical Considerations." **Strategic Management Journal,** 22**,** 875-888.

Priem, R. L. & Butler, J. E. (2001). "Is the Resource-Based 'View' a Useful Perspective for Strategic Management Research?" **The Academy of Management Review,** 26**,** 22-40.

Priem, R. L. & Butler, J. E. (2002). "Tautology in the Resource-Based View and the Implications of Externally Determined Resource Value: Further Comments." **The Academy of Management Review,** 26**,** 57-66.

Rubin, J. (1994). ***Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests***. New York: John Wiley & Sons, Inc.

Schoemaker, P. J. H. (2002). ***Profiting from Uncertainty: Strategies for Succeeding No Matter What the Future Brings***. New York: Free Press.

Schulze, W. S. (1994). "The two schools of thought in resource-based theory: Definitions and implications for research." **Advances in Strategic Management,** 10A**,** 127-151.

Simon, H. A. (1998). "Discovering Explanations." **Minds and Machines,** 8**,** 7-37.

Sowa, J. L. & Dietz, D. (1999). ***Knowledge Representation: Logical, Philosophical, and Computational Foundations***. Pacific Grove: Brooks/Cole Publishing Inc.

Spencer, L. (1989). ***Winning Through Participation: Meeting the Challenge of Corportate Change with the Technology of Participation.*** Dubuque, Iowa: Kendall/Hunt Publishing Company.

Swarm Development Group (2003). "SWARM." [Computer software].

Sycara, K., Decker, K., Pannu, A., Williamson, M., & Zeng, D. (1996). "Distributed Intelligent Agents." http://www.cs.cmu.edu/~softagents/ [On-line]. Available: **http://www.cs.cmu.edu/~softagents/**

Tapscott, D. (2002). "Rethinking Strategy in a Networked World (or Why Michael Porter is Wrong about the Internet)." **Strategy and Business,** 24.

Teece, D. J., Pisano, G., & Shuen, A. (1997). "Dynamic Capabilities and Strategic Management." **Strategic Management Journal,** 18**,** 509-533.

Ugwu, O. O., Anumba, C. J., & Thorpe, A. (2001). "Ontology development for agent-based collaborative design." **Engineering Construction & Architectural Management,** 8**,** 211-224.

University of Chicago: Social Science Research Computing (2002). "RePast 2.0." [Computer software].

Washbish, J. (2003, November 1). "Confessions of a Technophobe." **Aftermarket Business, November,** 44-52.

Wilson, P. H., Harnish, K., & Wright, J. (2003). ***The Facilitative Way***. Shawnee Mission, KS: TeamTech Inc.

Wooldridge, M., Jennings, N. R., & Kinny, D. (2000). "The Gaia Methodology for Agent-Oriented Analysis and Design." **Autonomous Agents and Multi-Agent Systems,** 3**,** 285-312.

Zimmerman, D. E. & Muraski, M. L. (1995). ***The Elements of Information Gathering***. Phoenix: Oryx.