Journal of the Association for Information Systems
JAIS

Special Issue

An Agent-Based Collaborative Approach to Graphing Causal Maps for Situation Formulation*

Douglas A. Druckenmiller
Department of Information Systems and Decision Sciences
Western Illinois University
DA-Druckenmiller@wiu.edu

William Acar
Graduate School of Management
Kent State University
wacar@kent.edu

Abstract

We provide a background discussion of group support systems (GSS) research into aiding strategic management processes. GSS support for strategic management has been primarily focused on qualitative analysis and the communication processes surrounding strategic planning. While fully developed in common decision-support systems, powerful simulation modeling and quantitative analytical tools have been difficult to integrate into GSS system configurations because they require increased cognitive load and expert modeling support, a central problem now addressed by collaboration engineering. A conceptual and functional bridge is needed to integrate the qualitative and quantitative approaches, reduce cognitive load, and provide modeling support that does not require experts. Acar's analytical causal mapping is introduced as a structured method for situational formulation and analysis of unstructured strategic problems. This form of causal mapping includes specific processes and analytical approaches offering cognitive modeling support for problem formulation. Its computational capabilities provide support for Systems Thinking approaches in a system easy to learn and use. Using the methodological template of the design science paradigm, we contribute a prototype system for the development and simulation of causal maps that uses RePast 2.0, a Java agent-based modeling (ABM) and simulation library.

Keywords: *Collaboration Engineering, GSS, Cognitive Decision Models, Strategic Decision Making, Artificial Intelligence, Modeling and Simulation.*

---

* Robert O Briggs, Gert-jan de Vreede and Anne Massey were the accepting editors.

Volume 10, Special Issue, pp. 221-251, March 2009

An Agent-Based Collaborative Approach to Graphing Causal Maps for Situation Formulation

1. Introduction

The potential for technological support of strategic management using group-support-systems **(GSS)** software has been the subject of research for many years (Tyran et al., 1992) and continues unabated, as evidenced by the feature article in the October 2006 issue of *ORMS Today* on computerizing multi-party negotiations (Kettelle, 2006). This body of research has primarily focused on communication processes and qualitative analysis, and has found that specific communication technologies have an interdependent relationship with organizational form, where the interplay between form and technology is an important aspect of organizational change. The integration of computer technology and communications capabilities has resulted in a richer communication capability for organizations (Fulk and DeSanctis, 1995). This richer communication capability has directly contributed to the success of the strategic planning process in many organizations. In a study involving 30 organizations that used GSS for strategic planning activities, researchers reported that: "The ability of the GSS to provide process support was found to be the most important contributor to strategic planning success...." (Dennis et al., 1997)

Despite the improvement in the communication processes of strategic management found by GSS research, the integration of quantitative approaches and tools for strategic management has been problematic. A conceptual and functional gap between qualitative and quantitative analytical techniques exists and is unaddressed in GSS research. A new paradigm needs to take hold. This paper discusses modeling support for strategy and policy using cognitive and causal mapping approaches.

The integration of qualitative and quantitative approaches compounds the problem of system adoption because this increases cognitive load and requires expert modeling support, a central concern of collaboration engineering. We approach this problem by designing a conceptual and a functional bridge that integrates the two approaches, reduces cognitive load, and provides modeling support that does not require quantitative expert facilitation. We introduce a new methodological avenue of Acar's (1983) *Comprehensive Situation Mapping* (**CSM**) as a causal mapping system, offering semantically rich modeling support for problem formulation that amounts to a new paradigm for scenario planning. We tap his analytical causal mapping as a structured method for situational formulation and analysis of unstructured strategic problems. It includes dialectical processes and analytical methods offering cognitive modeling support for problem formulation, and its computational capabilities provide support for Systems Thinking approaches in a system that is easy to learn and use.

2. Background

The development of information technologies for quantitative analysis has been primarily focused on structured analytical techniques (Tyran et al., 1992) including System Dynamics modeling (Forrester, 1961) and other powerful simulation and modeling tools. Strategists, however, have long been suspicious of computer modeling and simulation despite the availability of powerful tools used by operations research and management science (**OR/MS**). 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 and Mitroff, 1981). In recognition of this fact, the main European operational research journals, the *Journal of the Operational Research Society* and the *European Journal of Operational Research*, normally dedicated to problem solving, regularly dedicate special issues to the alternative path provided by problem-structuring methods (**PSMs**). Recent reviews of these alternative procedures by Franco (2006) and White (2006) show them to be multiple, divergent, and often over-compensating for the formalism of OR/MS by espousing primarily softer, more qualitative approaches.

Researchers in System Dynamics modeling (Andersen et al., 2006; Bryson et al., 2004; Andersen et al., 1997; Vennix et al., 1997) and cognitive mapping (Andersen et al., 2006; Ackermann and Eden, 2005) have been working together on finding the right mix between methodological power and richness with a "Systems Thinking" approach. In explaining the difference between System Dynamics and Systems Thinking, Richmond (1994) identifies three impediments to representing complex mental models as the series of stocks and flows central to System Dynamics modeling:

1. The gap between the explicit graphical representation and the tacit mental model;
2. The complexity of the explicit model that results in cognitive overload;
3. The ambiguity of the semantics of System Dynamics diagramming.

As a solution to these problems, Richmond proposes a higher-level mapping language that can be translated into the more detailed stock-and-flow diagrams of traditional System Dynamics to help bridge the gap between the user's mental model and the explicit representation in the model. Richmond points out:

> …one of the major "problems" with System Dynamics was the "we have a way to get the wisdom, we'll get it, then we'll share it with you" orientation. I feel that Systems Thinking should be about helping to build people's capacity for generating wisdom for themselves [(1994), p. 12].

Given the experience of GSS research into the problem of self-extinguishing GSS implementations (Briggs et al., 2003), this does not bode well for System Dynamics adoption. The adoption and diffusion research has focused on the issues of slow adoption of GSS in organizations and project teams, and on factors influencing sustained use of GSS (Dennis and Garfield, 2003; Briggs et al., 2003).

While the focus in much of collaboration engineering is on user-friendly packaging of facilitation elements, the same applies to the software that is part of the GSS system (Briggs et al., 2003). Also, because the cognitive load imposed by technological facilitation of complex qualitative processes is substantially increased if quantitative modeling and simulation tools are added to the mix, they have been relegated to more tactical use in the back office than use by key decision makers in the front office. Consequently, minimizing cognitive load for practitioners is a key goal of adoption and diffusion research that focuses on the lack of sustained GSS use (Dennis and Garfield, 2003; Briggs et al., 2003). The benefits of quantitative modeling must outweigh the cost of the increase in cognitive load to achieve sustained use.

The benefits of GSS support for cognitive mapping are well established and researched (Chaib-draa, 2002; Heintz and Acar, 1994), and Eden's work in this domain (Eden and Ackermann, 1998) is considerable. These tools can effectively mitigate problems associated with managerial cognition. Research (Hodgkinson et al., 1999) also shows that collaborative causal mapping reduces cognitive bias in the strategic decision-making process. Electronic brainstorming research (Potter and Balthazard, 2004) using a cause-cueing technique, which directs attention to the underlying causes of problems identified by participants, shows that such direction increases the quality and quantity of generated ideas.

This paper specifically contributes a prototype system for the development and simulation of causal maps using RePast 2.0, a Java-based ABM simulation library. Our research uniquely combines cognitive approaches to strategic thinking, knowledge management and agent-based simulation in a prototype system for situation formulation in a group decision-making environment. It directly addresses the cognitive load problems identified by Richmond (1994) and is a unique approach to the current Systems Thinking. The evaluation and testing of the prototype shows that it is easy to learn and use (Druckenmiller et al., 2007). The benefits gained from causal mapping and the reduction of cognitive load in a simplified tool potentially offset the added costs of quantitative modeling and simulation for key decision makers.

We also argue that this research contributes to collaboration engineering research in ThinkLet design and construction for the organizing pattern of interaction. While the research reported here focuses on the tool development side of ThinkLet construction, the overall theory of problem formulation also has a procedural base in dialectical inquiry that goes hand in hand with the development of the tool. We are currently testing facilitation scripts with usability testing approaches. This research shows how usability testing and the *design science* method are fundamental to innovative ThinkLet development. The research also contributes to the distributed artificial intelligence literature, because of the innovative use of agent-based modeling and simulation for conceptual modeling, in addition to established environmental and social modeling applications.

The paper first discusses the methodological context for this research, which follows the design-science paradigm (Gregor and Jones, 2007; Hevner et al., 2004; Carlsson, 2006; Vahidov, 2006). We next provide a brief discussion of the theoretical framework that provides for the rigorous development of a tool to address the problem. Then we discuss the development of a testable software prototype that is the primary contribution of the research. In discussing the development of the prototype, we present the design search process, the current prototype design, an example of the software's forward analysis capability, and a brief summarization of the design's evaluation through usability testing. We finish the paper with a brief discussion of future research developments and some concluding statements.

3. Methodological Context: The Design-Science Paradigm

For its methodological backbone, the research reported in this article utilizes the design-science paradigm (Gregor and Jones, 2007; Hevner et al., 2004; Carlsson, 2006; Vahidov, 2006). This paradigm is an accepted approach to research and is complementary to the behavioral science paradigm rooted in behavioral science research. While behavioral science research seeks to develop and justify theory, design science seeks to solve problems through innovative and creative development of artifacts that apply, and test, kernel theories and methods through an iterative process of development (Kuhn, 1996; Simon, 1996; Berente and Lyytinen, 2006). Hevner et al. (2004) provide guidelines for design science research. This article addresses these guidelines in the following ways:

- **Research Communication**: For collaboration engineers, this research focuses on the development of a new tool for construction of ThinkLets in the organizing pattern of interaction. Our research describes an easy-to-use tool for Systems Thinking and conceptual modeling of problem situations. For researchers, it provides an excellent example of the design-science paradigm applied to developmental collaboration engineering research. And for practitioners, it offers a description of an easy-to-use software tool for practical use in consulting situations involving group model building and systems thinking.
- **Research rigor**: The next section gives a theoretical perspective on a general theory of problem formulation though situational analysis based on CSM causal mapping. This research has been instantiated in a prototype system now tested through the accepted usability testing procedures described in the usability testing literature. While the test plan and procedures have been reported elsewhere in the literature (Druckenmiller et al., 2007), we summarize the test results in this paper as evidence of the evaluation of the research through testing. We also argue that usability testing has an integral role in design science research and shares much of its goals and procedural base.
- **Problem relevance**: This research proposes a solution to the problem raised by Richmond (1994) and the Systems Thinking movement in their addressing the facilitation and cognitive load problem of System Dynamics modeling. In an effort to enable adoption of the Systems Thinking approach in organizations, we elaborate and test a theory of problem formulation and framing that allows practitioners to develop Systems Thinking models themselves.
- **Design as a search process and an artifact**: The body of this paper describes the *development of a CSM tool* that instantiates a general theory of problem framing in a prototype software tool. The objective was to create a computerized device that accomplishes the primary core of CSM's "backward" and "forward" analysis capability for full situation analysis, and yet is easy for the general user to learn and use. We describe several iterations of the *generate-test* cycle, the problems encountered and solutions found in the iterative design search process.
- **Design evaluation**: The paper also summarizes usability testing results. We conduct usability testing of the initial prototype according to standard procedures described in the usability testing literature. The details of these tests have been reported elsewhere in the literature (Druckenmiller et al., 2007); still, we summarize the findings in this article following the description of the development of the software.

- **Research contribution**: This research makes several contributions. First is the software prototype itself. Hevner et al. (2004) note that, in the design-science paradigm, the *artifact* (in this case our CSM Tool) is *itself* a research contribution. In short, we are contributing a new tool for group model building and Systems Thinking, based on an innovative theory of general problem formulation and specifically aimed at Richmond's (1994) problem. Additionally, the research contributes to understanding the role of design science and usability testing as tools for ThinkLet construction; this offers a new avenue for agent-based modeling and simulation tools.

4. Analytical Situation Framing or Problem Structuring

4.1 Using Systems Thinking by Design

Design science attempts to address the gap in Information Systems research between the rigor of theoretical formulation and the relevant practice embedded in the immediate need of business situations (Carlsson, 2006). The method we advocate is being developed as a bridge between the rigorous, modern, solutions-oriented quantitative methods of OR/MS and the relevant, qualitative, problem formulation approach of process consulting (Acar, 1987). Critical to any problem-solving situation is the formulation of the problem. The strengths of OR/MS approaches lie in their quantitative solutions to formulated problems. The difficulty, hence, is no longer in solving problems using quantitative methods, but in framing or formulating a problem situation to which various OR methods could be applied, as illustrated in the "bridge" of Figure 1.

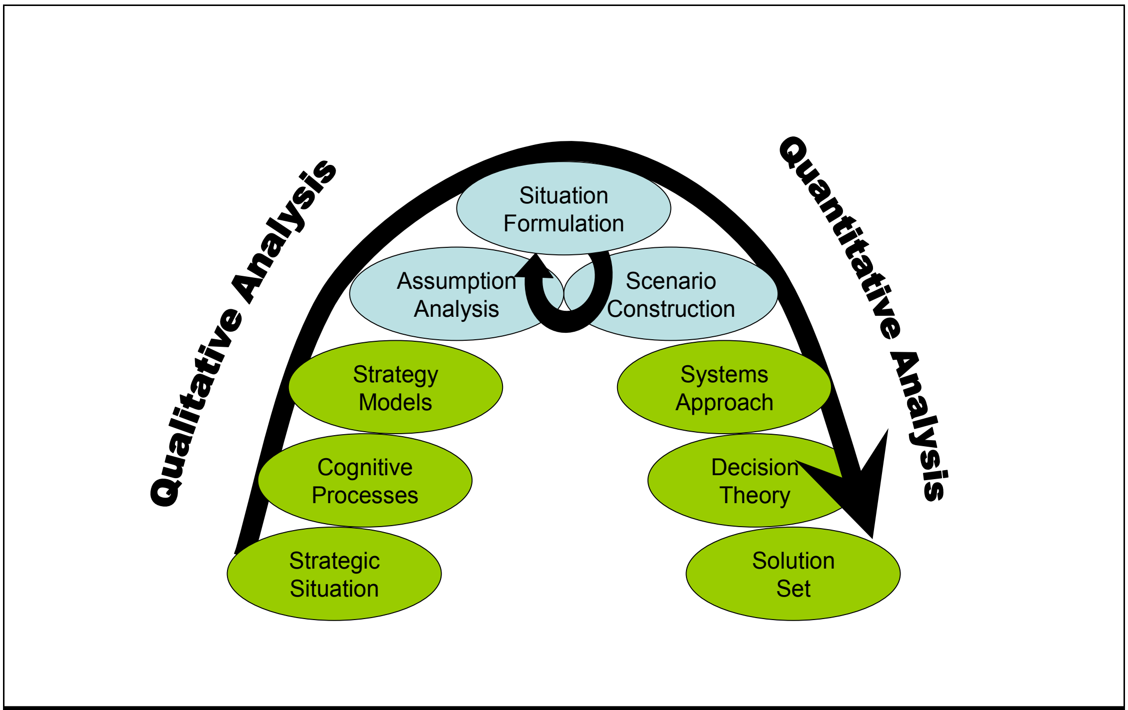


Figure 1. Acar's CSM bridge (Acar, 1987)

In this regard, two research thrusts have made a seminal contribution to turning problem formulation into a science. On the *theoretical* front, the work of Churchman (1971) and his colleagues, such as Ackoff (1981) and Mason and Mitroff (1981), is increasingly viewed as providing a philosophical foundation for the design of inquiring systems and continues to be cited (Peachey and Hall, 2006). Specifically, building on his analysis of the design of inquiring systems, Churchman contributes to the systems approach an interactive search process that he termed **dialectical inquiry.** Mason and Mitroff start with the realization that little attention has been traditionally paid to the surfacing and testing of strategic assumptions in GSS research. Contemporary research confirms these insights, in that it usually finds that idea generation with GSS that uses assumption reversal techniques in electronic brainstorming promotes creativity and the production of fewer but better ideas (Hender et al., 2002).

On the *methodological* front, real-world problems and policy situations that exhibit complexity have to be viewed as composed of many interrelated problems and issues. Uncertainty and turbulence in the environment, competition, firm capabilities and implementation tactics necessitate a comprehensive approach to strategic problem formulation (Georgantzas and Acar, 1995). Problem *framing* (or *structuring* as denoted earlier) is gradually replacing traditional problem solving (Checkland, 1981) in larger-scale interventions. A systems approach to problem formulation stresses that isolated problems cannot be isolated from surrounding messy realities. Therefore, framing the whole situation to understand its complex dynamics is central to finding holistic solutions.

The messiness of reality requires a shift from formulating single problems to formulating entire situations (Ackoff, 1981). And, to be effective, strategies must holistically address the complexity of the entire situation rather than propose solutions to single problems. “Situation formulation” (Acar, 1983) is an approach aimed at the integration of qualitative and quantitative analytical techniques in a single conceptual modeling approach that graphically represents strategic knowledge and is open to computation. This holistic approach is compatible with the recent view of Knudsen and Levinthal (2007), who hold that choice sets are not available ex ante to actors, but must be constructed.

The systems approach to situation formulation has generated a variety of strategic knowledge representation techniques. Graphic representations are well known in the management and social science literature as both systems analysis tools and knowledge representation techniques (Sowa, 1999). Huff and Jenkins (2002) identify a variety of cognitive mapping techniques derived from cognitive psychology. More broadly, in addition to the *personal construct theory* of cognitive psychology (Kelly, 1955), a number of other approaches have been developed, including adaptations of mathematical *graph theory* (Harary et al., 1965), *influence diagramming* (Maruyama, 1963), *causal mapping* (Acar, 1983), and *systems dynamics* (Forrester, 1961).

On the *operational* side, the abundant work of Eden and his team, e.g., Eden, 1990; Eden and Ackermann, 1998; Bryson et al., 2004; Ackermann and Eden, 2005, spanning three decades—first at Bath then at Strathclyde—and countless examples, has provided a wealth of details on ways to undertake the initial steps of problem formulation. In fact, Ackermann and Eden’s recent book (2005) is aptly titled: “*The Practice of Making Strategy: A Step-by-Step Guide.”* Their patient work documents how, in numerous and diverse situations, managers’ intuitive synoptic views can be captured by a variety of cognitive diagrams that support enhanced reflection and communication. This work in cognitive mapping is trail-blazing in the area most neglected by the powerful OR/MS battery of tools, the front-end area addressing the **framing** or structuring of the problem(s). Once this is done, Eden and his coauthors patiently guide their managerial advisees to benefiting from the power of full-fledged computer simulation, often using some version of the powerful System Dynamics software.

Can the already rich thrusts of the Churchman and Eden schools be somewhat combined? Acar (1983) has proposed combining dialectical inquiry with an analytical causal mapping technique; for this reason, it is called *Comprehensive Situation Mapping* (CSM) (Acar and Druckenmiller, 2006). CSM is both a collaborative process and an analytical framework that can be used for surfacing assumptions as well as devising strategic scenarios to aid the development of organizational foresight. Scenarios are crucial in dealing with “wicked problems” (Mason and Mitroff, 1981). The analysis of **change scenarios** (Schoemaker, 2002; van der Heijden, 1996) allows the design of strategies to take place in spite of the messiness of the typical situation.

Thus, CSM represents an integrated approach that combines the rigor of OR/MS modeling with group process technologies supporting situational analysis and problem formulation (Heintz and Acar, 1994). Scenario-driven planning is a holistic approach to situation formulation and strategy development. It is accomplished by blending qualitative planning processes based on assumption surfacing with quantitative modeling and simulation in a unified methodological framework (Georgantzas and Acar, 1995). As a process, CSM can, thus, be viewed as a collaborative dialectical conversation that, through negotiation, develops a common conceptualization (causal map) of the problem situation to be addressed.

4.2. The CSM Analytical Technique

We chose to work with Acar's CSM approach because it differs from other approaches using weighted links. For example, let us consider the fairly typical method recently proposed by Scavarda et al. (2006) for constructing a collective causal map. Although new visualization software (such as Bouzdine-Chameeva's ANCOM-2) was implied, the links are simply evaluated on a 7-point Likert scale as an assessment of the perceived strength of the relationships. Let us also examine less common procedures, such as the ones recently reviewed by Montibeller and Belton (2006). In an original method of their devising, the nodes of the causal graph are laid out vertically, with the bottom nodes indicating attributes, the middle ones representing consequences, and the top nodes denoting value concepts. In addition, they allow for qualitative or quantitative assessments of the strengths of links as well as the signs of the relationships, thus allowing, in some cases, computation of the algebraic strength of the presumed relationships. On the other hand, Acar's CSM (1983) is of interest here because it is geared toward endowing cognitive mapping with computational properties with respect to calculating, not just the strength of the links, but *the propagation of change* in a causal network. This is of an altogether different stripe from extant methods, as it opens the door to the computation of **change scenarios**.

While the specific semantics and techniques of CSM have already been published in the literature (Acar and Druckenmiller, 2006), we present them here briefly for the benefit of the reader. Figures 2 and 3 summarize the technique and its semantics. In contrast with most varieties of cognitive mapping, whose procedures vary from one sub-technique to the next, the node elements of a CSM causal graph always represent *quantifiable* variables. Other restrictions militate to make CSM a methodical technique, one of which is that a strict interpretation applies to each type of link connecting the nodes displayed in the example of Figure 2. CSM causal relationships are modeled through three channel types:

- double arrows or *full channel* expresses an unfettered capability to transmit change from an upstream node variable to a downstream one;
- a single arrow or *half-channel* expresses that several upstream links have to be activated together for change to be transmitted to a downstream variable;
- a dashed arrow expresses a *restriction* or constraint on the transmission of change from node to node.

In addition, algebraic change-transfer coefficients or *transmittances* indicate the sign and magnitude of change (for example, the notations "+.50" in Figure 2 to indicate a proportion of ½ between the change received from an upstream node and the resulting change in the downstream one). Also indicated are the minimum thresholds, if any, and *time lags* (for example, the notations "2m" to signify time lags of two months between reception of a change signal and the actual transmission of the change).

This somewhat complex notation affords CSM a mathematical underpinning. Among all the easily used graphical procedures, CSM is the one to date to identify the *intensity of transmittance* of a causal link in a graph to the derivative (or slope) of a linear relationship among the downstream and upstream node variables. Figure 3 summarizes this by showing how a linear relationship between a downstream variable Y and an upstream one X translates into a proportion between an increment in X and the resulting one in Y, the *magnitude* of which is expressed by the "transmittance" or change-transfer coefficient of the channel.

Treating change increments as mathematical differentials drops the constant terms, and Acar (1983) developed a technique for computing the resulting changes at each successive downstream node in a causal network. He showed the resulting process of the cascading propagation of change to be *transitive* within chains of full channels. CSM also makes a provision for flagging on the graph the sources from which change emanates, and marking up the *goal* vectors next to the focal nodes of the network. The flow of change through the system is, thereby, quantified and may be used for in-depth analyses.

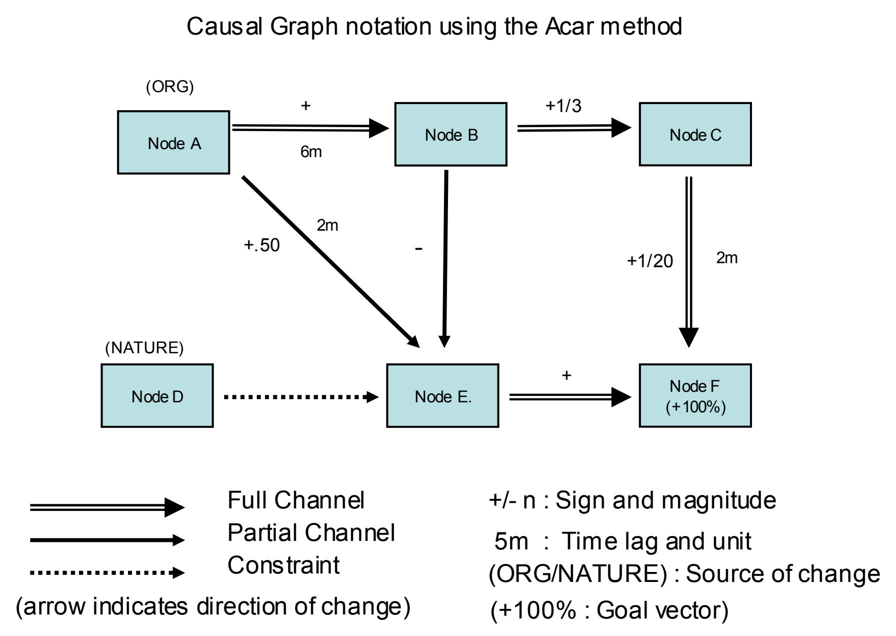


Figure 2. An Example of CSM (Acar, 1983)

CAUSAL LINK SEMANTICS

Change Transfer & Time Lag Coefficients

$$X \xrightarrow[\Delta t = k]{a} Y$$

Implies

$$Y_{t+k} = a X_t + b$$

OR

$$\Delta Y_{t+k} = a (\Delta X)_t$$

NOTATION

a : change-transfer or "transmittance" coefficient
K : time lag

Figure 3. Link coefficient notation - mathematical basis.

The preceding summarization of CSM demonstrates the rigorous theoretical base of the current development and how it addresses the theoretical gap that exists between quantitative and qualitative approaches to problem formulation. As such, it represents a general theory of problem formulation and represents the rigorous theoretical base of this research. But theoretical solutions must also be relevant. The following section addresses the consulting process that facilitates the construction of causal maps as knowledge artifacts that represent a formulated problem or framed situation.

4.3. Using CSM in a Consulting Process

As a process-oriented method, Acar's CSM progresses through several stages or "levels" in the development of causal graphs (Acar, 1983). From a collaboration engineering perspective, the CSM process starts with *divergence* among the participants. In the initial Survey or "Level 0" stage, the various participants create individual causal maps by identifying the key actors and variables in the system of interest. This cautious beginning is followed, in the "Level 1" Structuring stage, by building a semantic network map. In the final Graphing or "Level 2" stage, the causal linkages are finalized on the individual maps and goal nodes are identified. Figure 4 gives a synoptic view of the entire CSM process and is particularly applicable to its initial divergent part.

This initial divergence is later complemented by a *convergence* part using *some or all* of the analytical components shown in Figure 4, in which a common causal map is collaboratively developed using the insights from the individual maps. This second, convergence part of the process creates at least partial consensus through dialectical inquiry in which all stages are revisited in a group context. Throughout the convergence and divergence parts of the CSM process, major and minor assumptions about the situation are surfaced and tested, resulting in commonly held perceptions of the situation expressed in maps similar in structure to Figure 2.

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 possible overlaps. Like other digraph-based approaches, Acar's (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 classical structural analyses of graph change, reachability, graph scope and connectivity can be initiated as early as the Level-0 Survey stage, but they would be more effectively conducted (and completed) at the Level-1 Structuring stage as illustrated in Figure 4. Similarly, goal analysis could be stated early or could await the Level-2 full Graphing stage.

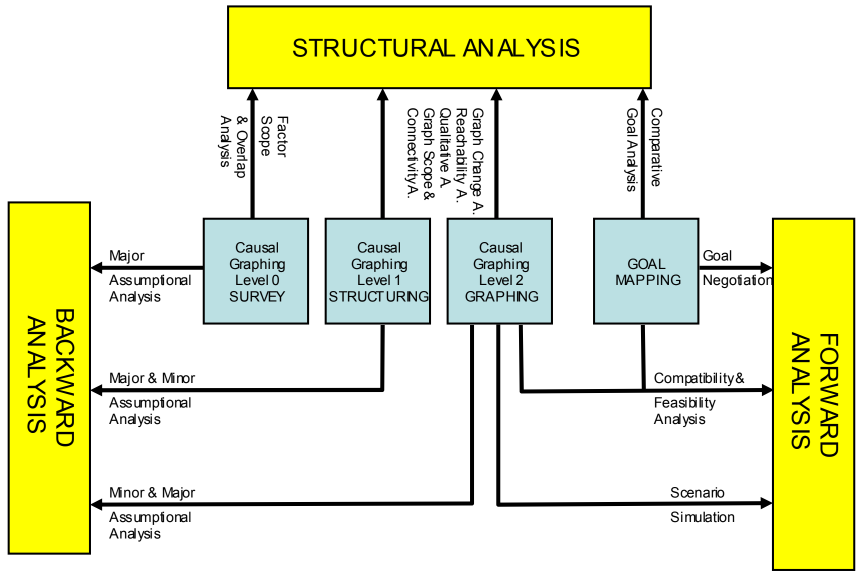


Figure 4. A synoptic view of Acar's system: components of causal mapping (Acar, 1983)

Backward Analysis

Churchman (1968; 1971) and Mason and Mitroff (1981) have brought to the fore the importance of creating, during strategy design, the conditions favorable to the surfacing of the (often hidden) underlying assumptions. For them, it is the surfacing and testing of implicit assumptions that should form the basis of the development of newer, more robust strategies. Yet little use is being made of their concept of "assumptional analysis," in spite of the finding that idea generation with GSS in electronic brainstorming promotes creativity when assumption reversal techniques are used (Hender et al., 2002).

In this vein, CSM stresses repeated reexamination of the assumptions underlying a group's causal map. At any time, but mostly during its Level-1 Structuring stage, it solicits comparisons among individual decision makers' (or decision-making groups') views of the situation, and of changes in those views over time. This affords us the "**backward**" **analysis** of assumptions advocated by systems theory authors. In other words, analysis of key assumptions about the problem situation's factors, variables, actors, and their relationships is undertaken to get a better sense of the qualifications and quantifications of those relationships (see Figure 4). By adding to the systems theory a technique based on a clearly identified causal graph to fix ideas, the backward analysis component of CSM is more effective than the free-debate Dialectical Inquiry process of Churchman (1971) and the "Strategic Assumption Surfacing and Testing" process of Mason and Mitroff (1981).

Forward Analysis

The Level-2 Graphing and Goal Mapping stage adds to situation formulation and structuring a forward-analysis capability. Its "**forward**" **analysis** component is what truly differentiates CSM from the competing approaches tallied by Huff (1990). This stage of the CSM method includes goal negotiation and analyses of goal compatibilities and feasibilities. More importantly, by including in the method indications, not only of the signs of the presumed causal influences but also of their intensities and possible time lags, Acar (1983) developed a technique for simulating manually (on the causal map itself) the propagation of change through a causal network. This computational capability raises CSM to a level beyond dialectical-only methods, because it endows its users with an ability to *compute* change scenarios.

Our present computerization of it, therefore, goes beyond alternative scenario-based approaches (Schoemaker, 2002; van der Heijden, 1996) in that it blends qualitative planning processes and assumption surfacing with quantitative modeling and simulation in a unified methodological thrust as recommended by Georgantzas and Acar (1995). Undertaken with CSM, scenario-driven planning becomes a holistic approach to situation framing and strategy development conducive to collaborative organizational learning.

Advantageous Features of CSM

In summary, because of its forward analysis capability, the full CSM method allows for goal mapping and development of scenarios that aid strategic planning and formulation. In addition, due to its backward analysis feature, the process proceeds in stages, first allowing for divergence among the participants and later guiding them toward convergence of their views. Because of its combination of analytical and dialectical features, we deem CSM to be a Singer-Churchman (Churchman, 1971) inquiring system because different perceptions of a situation are merged through a "sweeping-in" process providing an inter-subjective view of the situation: the causal-graph based negotiations create a foundational consensus that forms the basis for the development of newer robust strategies. These features of the CSM method gave us a foundation on which to build a state-of-the-art GSS prototype, and thus a means for implementing the emerging paradigm of *collaboration engineering*.

5. Design Search Process and Artifact for a Testable CSM Prototype

5.1 Early Design Prototypes and Testing

W. Acar's (1983) CSM causal mapping approach was designed in the early 1980s as a manual system because, at the time, routinized systems could not support much interactivity. Later work attempted to computerize the various component analyses of the CSM method. W. Acar and T. Heintz (1994) showed how its collaborative aspect of map building and assumption analysis could be realized through object-oriented programming, and M. A. Acar (2000) illustrated the programming simplification that Java could bring to the task of computerizing the Survey and Structuring stages.

While object-oriented approaches to CSM automation initially appeared promising, difficulties in simulating complex causal loops led to significant technical limitations. Implemented as a SmallTalk application, that early prototype demonstrated the potential of object-oriented techniques, but was limited to the graphic editing of causal maps and could not implement CSM's Forward Analysis feature. Our initial prototype study suggested a solution based on distributed artificial intelligence (**DAI**); yet it did not point to any specific implementation mode. This constituted a serious limitation of our initial prototype and precluded its use for scenario development.

The recent development of agent-based modeling (**ABM**) and simulation tools, e.g. Zhiang et al., (2006), provides a distributed solution to forward analysis. Thus, the major challenge addressed at this stage of development was to enable the system to computerize the *forward analysis of causal maps.* This was by no means a trivial challenge. The complexity of causal loops in the most causal graphs entail computing, at each node, the cumulative impact of successive waves of change through the model when the propagation of change in the causal network is simulated. This presented a substantial design challenge for developing automated scenario support using object-oriented techniques alone.

5.2. Agent-Based Modeling (ABM) Solutions

ABM simulation systems have only recently been used to research and analyze business systems and environments (Robertson, 2003; Zhiang et al., 2006). Originally applied to biological and ecological contexts for modeling complex adaptive systems, the technique is now being applied to the social sciences as well. Embraced by a growing number of scientific researchers in a variety of natural and social science disciplines, agent-based modeling creates artificial worlds that model real-world environments. Automated agents are used to populate these worlds and, based on simple rules, simulate the behavior of their real world counterparts. Researchers use these simulated worlds to test theoretical and empirical constructs (Panepento, 2000).

Strategy is ultimately concerned with designing complex systems. ABM provides 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. Causal maps are models of such environments. Simulation of complex systems allows evaluation and observation of global behaviors and system dynamics that cannot be analytically predicted *ex ante,* and hence, can assist in the design of new complex systems composed of multiple interacting actors.

The interaction of autonomous entities in a common environment is typically a mutually recursive process that could become analytically intractable. ABM tools share with other computerized approaches a vulnerability to modeling and simulation: they are not user-friendly, are time consuming, 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. Aware of these traps, our current prototype offers the coordination of autonomous intelligent-agent activities with human control through agent-based conceptual models of causality. Developments in DAI and multi-agent systems offer powerful and integrated solutions to both problems outlined above. The next sections describe how agent-based development platforms allowed us to create a networked, platform-independent application that we linked with ABM simulation tools for generating the scenarios implied by the CSM maps.

The outcome of the design process is to produce an intermediate-level abstraction amenable to being 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 multi-agent systems for development is that a simulated system environment can be created to test how the system works in a variety of *controllable* test situations. 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 implement and develop the “forward analysis” capabilities of CSM. We investigated two agent-based modeling tools for potential use as an implementation environment for CSM: Swarm and Repast 2.0.

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* as a research tool for a variety of disciplines. 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 as a simulation system using collections of concurrently interacting agents. 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).

The conceptual basis of Swarm comes from the field of Artificial Life, which 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. The *Swarm Development Group*, a non-profit organization, supports the development of the Swarm Simulation System and its use by a community of researchers in a variety of agent-based models. Currently, Swarm is freely distributed for use under public license as a set of libraries that facilitate implementation of agent-based models.

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. Citing the University of Chicago manual that presents it:

> 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 that changes the state machine through a schedule. RePast moves beyond the representation of agents as discrete, self-contained entities and virus 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).

Therefore, RePast is 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,[1] 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).

5.3. Agent-Based Simulation of Causal Maps

The power of this application is in its ability to first develop specific environmental topologies using a causal mapping approach, and then simulate the strategic landscape with an agent-based modeling and simulation tool. The current prototype provides a graphic user interface for creating causal maps using Acar's semantic system. Once constructed, maps can then be put into motion (simulated) using RePast. Agent-based modeling and simulation 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 strictly object oriented approach to design, but RePast provides a robust simulation environment. Our simulation system is integrated into the editing interface of the application, and our linked editing interface provides an easily used graphic model-building tool for RePast simulations. With this novel approach, the *forward* analysis of causal maps (i.e., the simulation of the propagation of change within the causal network) can be done at any time to test the implications of created maps.

6. Current Design and Application Prototype

6.1. User Interface Design

We programmed the user interface 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 commonly used guidelines for user interface (Sun Microsystems, 1999) in Java application development. These guidelines provide for consistent interfaces that minimize user learning curves and give the application an "intuitive" feel. Bar menus, pop-up menus, dialogue boxes, etc., all provide standard windowed interface components. In designing the user interface, we followed previous developments. An early prototype (2000) implemented a Java based collaborative tool that provided a network-based application for sharing and editing maps; unfortunately, this initial prototype focused on the development of a client server application with minimal user interface development and no support for the forward analysis of causal maps. The user interface components are presented in Appendix A.

6.2. Agent-Based Modeling & Forward Analysis Simulation

In order to illustrate the forward analysis capability of the software, we constructed two illustrative maps of topical interest.

A Simple Topical Example

The first of these two maps represents a hypothetical 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 5 to 8 show such a hypothetical scenario developed for demonstration purposes. To better illustrate the method, we construct an alternate conceptualization of the same situation – that of an economist applying standard supply and demand theory–for comparison purposes and will discuss it in connection with Figure 9. The underlying assumptions of these two maps differ only in the directionality of one of the causal links as explained below. The simulations illustrate how a simple change of assumptions may radically alter the implications of a causal graph. These implications are difficult to anticipate from just interviewing decision makers or looking at their maps, but are dramatically revealed when simulated with the software.

---

[1] A micro-world environment that is basically a two dimensional grid can "wrap" in any direction. For example, should agents wander off the left edge of the world, they would appear on the right hand side of the grid; this is viewed as a *torus implementation*.

In the first example map (Figure 5), increased terrorist threats impact both consumer demand and flight availability, as flights are canceled. The link notation signifies that increased terrorist threats are assumed to cause a decrease of 5 percent in available flights and a decrease of 25 percent 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; this affects consumer demand. Decreased consumer demand means fewer people will travel to Europe. Fewer people travelling to Europe makes threats to air travel less attractive to terrorists, decreasing the threat level. Thus, this small CSM map represents a potentially disruptive terrorist's possible conceptualization of the situational links in a key sector in the European economy.

Computing the Forward Analysis

We construct a scenario around this situation with an increase of 10 percent in terrorist threats being the only trigger (see Figure 5). This is accomplished in our prototype 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 triggering change in the node from its *status quo* level of zero. **We also choose three goal nodes** (color-coded in crimson) 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 time plot (see Figure 8) showing the status of each goal node is displayed. The time plot 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.

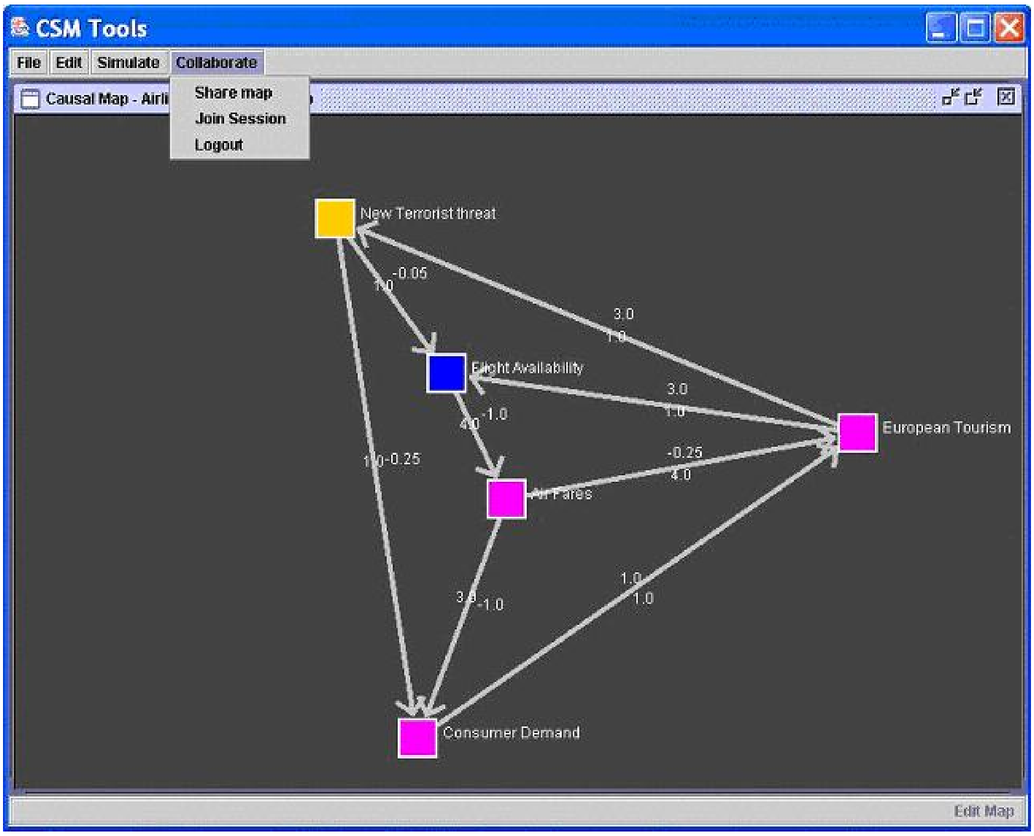


Figure 5. Editing interface

The link coefficients and time lags indicated on the CSM map of this example are for illustration purposes only and not based on empirical studies. Empirical validation of simulation models is an outcome of the process. At an early stage of map development, the coefficients and parameters to be entered are simply estimated by the decision participants when not easily retrievable from the firm's database. Since this places limitations on causal mapping, a primary future thrust of this research would be to enable agent-based validation of quantifiable assumptions in the model through data mining or other data analytical methods.

Once the scenario is constructed, we run an **agent-based simulation**. 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. 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, tabulating the calculations by hand would have become quite tedious because of the differing time lags associated with each link.

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.

Fortunately, our ABM solution allows the prototype to function smoothly. At tick zero, only the Terrorist Threats node has changed with the initial trigger value. All other nodes remain at their status quo level of 0. As the simulation progresses to time tick one, both the Flight Availability node and Consumer Demand nodes change value. Flight Availability has decreased to -0.5%; this value is obtained by multiplying the change in terrorist threat level (10%) by the link coefficient of -0.05.

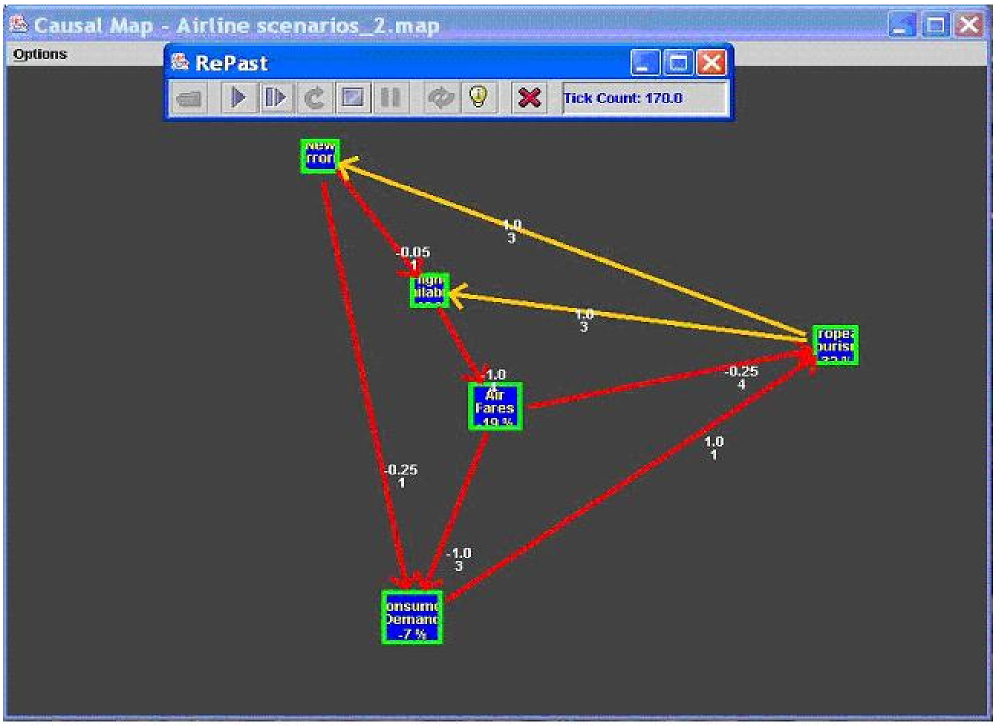


Figure 6. Repast simulation of edited map – simulation time 170 weeks

As the simulation progresses, the changes cascade through the map according to the time lags of the individual links. By tick five, the 2.5 percent decrease in European tourism that occurred at tick two has reduced the terrorist threat level by the same amount to 7.5 percent and has reduced flight availability by 3 percent Consumer demand and European tourism remain unchanged. 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 6 and Figure 7. The next section explains the meaning of the color-coding of the links and the variations in node size displayed in these figures.

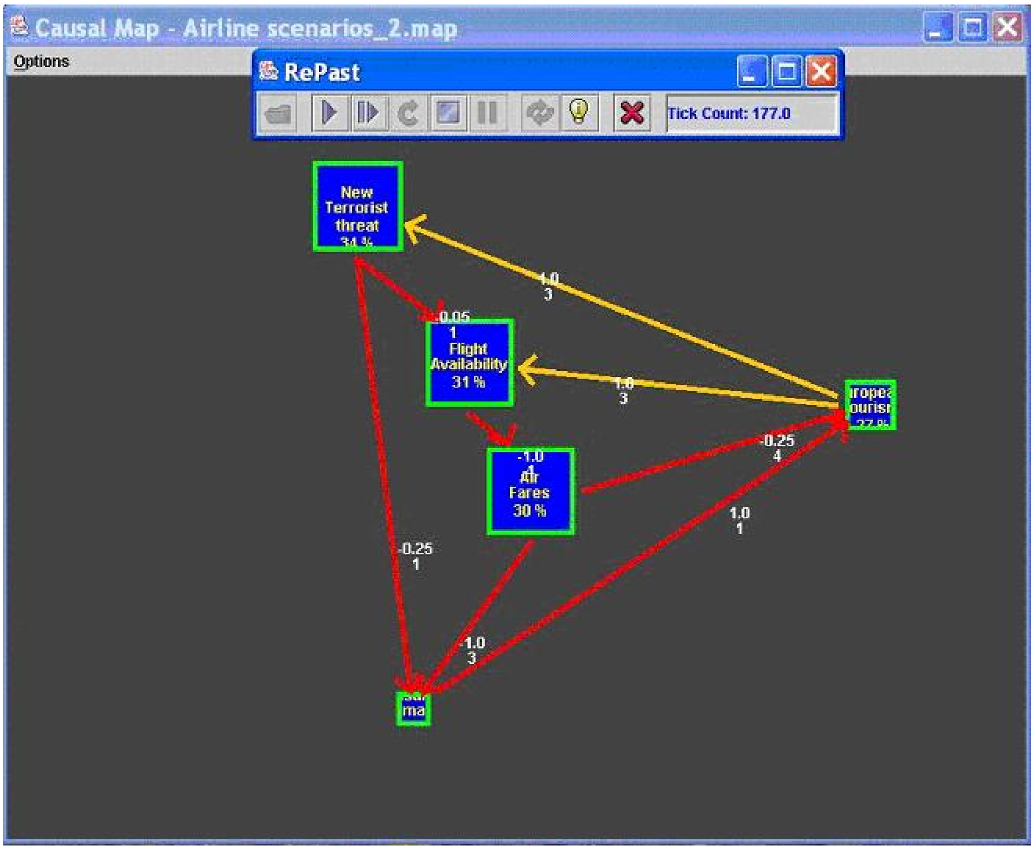


Figure 7. Repast simulation of edited map – simulation time 177 weeks

Visualizing the Forward Analysis

In our prototype, 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 the oscillations in Figure 8). 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 made our computerized causal mapping an engaging, dynamic, and interactive support tool for strategic decision support.

We color-code the propagation of change through the network to enable visualization. Figure 6 shows the terrorist scenario at 170 weeks of simulated time. The initial point of change in this scenario was an initial increase of 10 percent 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. Figure 7 shows the map at 177 weeks of simulated time. Terrorist threats have once again increased as available flights increase due to increased demand. More available flights provide renewed rich targets of opportunity, etc.

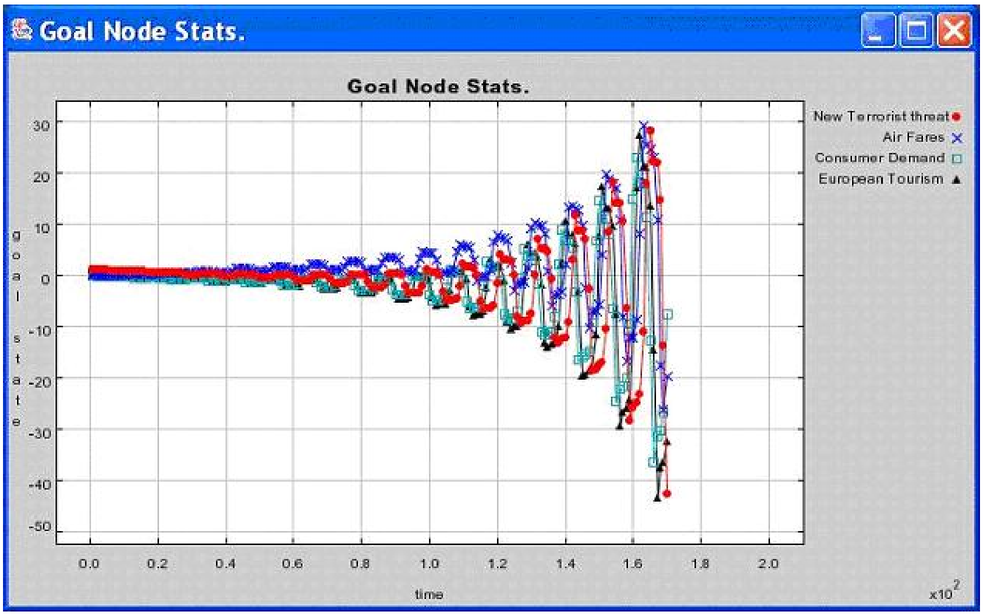


Figure 8. Repast simulation of edited map – node graph

Our prototype's output is more visual and easier to follow than existing applications of systemic concepts. The nodes are either *visibly increased or reduced in size,* and the current change from the *status quo* is indicated inside the node as a percentage. Our prototype also displays graphical summaries: Figure 8 is an *animated* graph of the oscillations of the levels of goal 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 oscillation captures visually the dramatically unstable scenario of economic destabilization caused by the terrorist of our example.

Interpreting the Forward Analysis

Because the simulated outputs are the implications of the situation as currently modeled, users can examine these outputs to help refine their assumptions and check the face validity of their model. In CSM, an iterative process of backward *assumption modeling* and forward *simulation analysis* represented in Figure 4 allows decision makers to understand the nature of the strategic situation and problem at hand. Alternate scenarios (due to changes in the initial conditions or changes in map structure) can be created and stored as separate maps.

For example, an analyst using economic theory as a guide would have a slightly different view of the above situation. Based on the basics of supply and demand, an economist would reverse the directionality of the link between Air Fares and Consumer Demand. He would point out to the modelers of the system that air fares are prices set by supply and demand, and that supply decreases should be offset by decreasing demand, resulting in equilibrium at a new price level. Figure 9 represents the economist's map and implicit scenario. The trigger node is still a change in the Terrorist Threats level and Air Fares are still selected as the goal node.

Figure 10 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, but is quite stable.

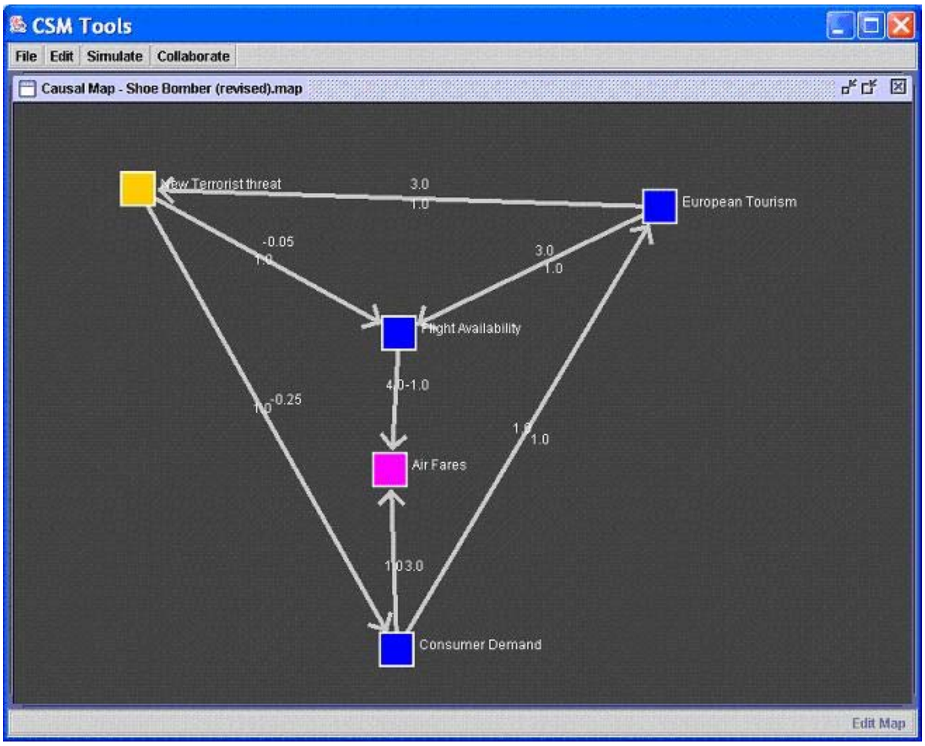
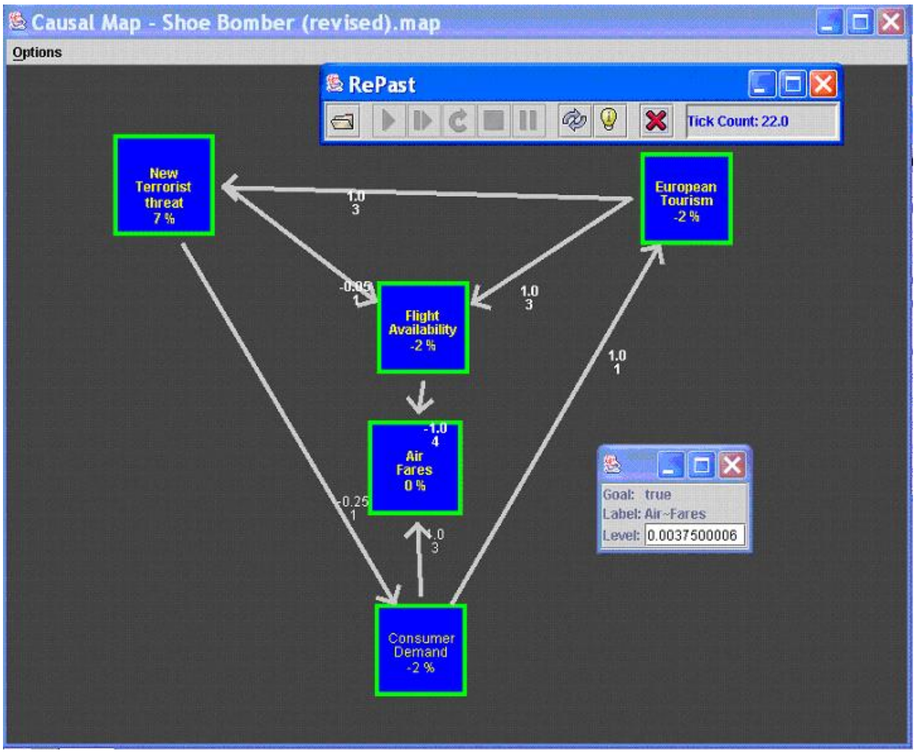


Figure 9. Revised example according to economic theory.



Goal: true
Label: Air~Fares
Level: 0.0037500006

Figure 10. Economist scenario at end of simulation.

7. Iterative Design: Implementation Problems and Solutions

Berente and Lyytinen (2006) identify the iterative nature of prototype development as a fundamental construct in information systems design. In the implementation of this prototype, several practical problems with theoretical implications had to be resolved. The development of the prototype is, thus, an evolutionary and incremental process. The construction of dynamic systems models and their implementation with an agent model required us to loop back to the theoretical fundamentals of CSM in order to find practical workable solutions. Since the constructed models are open-ended (they can specify an infinite variety of complex interactions between agent nodes), the scheduling of agent interactions and the possibility of infinite causal loops had to be specifically a part of the design. The next section discusses the nature of the problems and the solutions that developed.

7.1. 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 computerized 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 its *status quo* level. Thus, changes are cumulative and, since link coefficients can be negative or positive, the current level of the agent will increase or decrease. Changes do *not* automatically affect all agents, however. Threshold values and restrictions are defined that allow the agent to block or drop changes from propagating through the system, but active changes are transmitted on outgoing communication links. Thus, agents are both interactive and autonomous.

Scheduling

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 links one after another. 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 need not be implemented as agents.

The different kinds of links also affect agent behavior. Agents will process *full channel* linkages if threshold values allow. *Partial channel* changes, 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 fully additive, in that the change values on each link are not always added together. In Acar’s (1983) system, these links model the semantics of the necessary or sufficient conditions for a change to take place (in CSM, partial link changes take the minimum value of the associated links). *Restriction links* 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.

Change is initiated in the system by the user’s specification of 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) or internal levers (internal agents that represent controlled agents in the system of interest.) These “sources of change” 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.

The Possibility of Infinite Loops

The node agents 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 11.

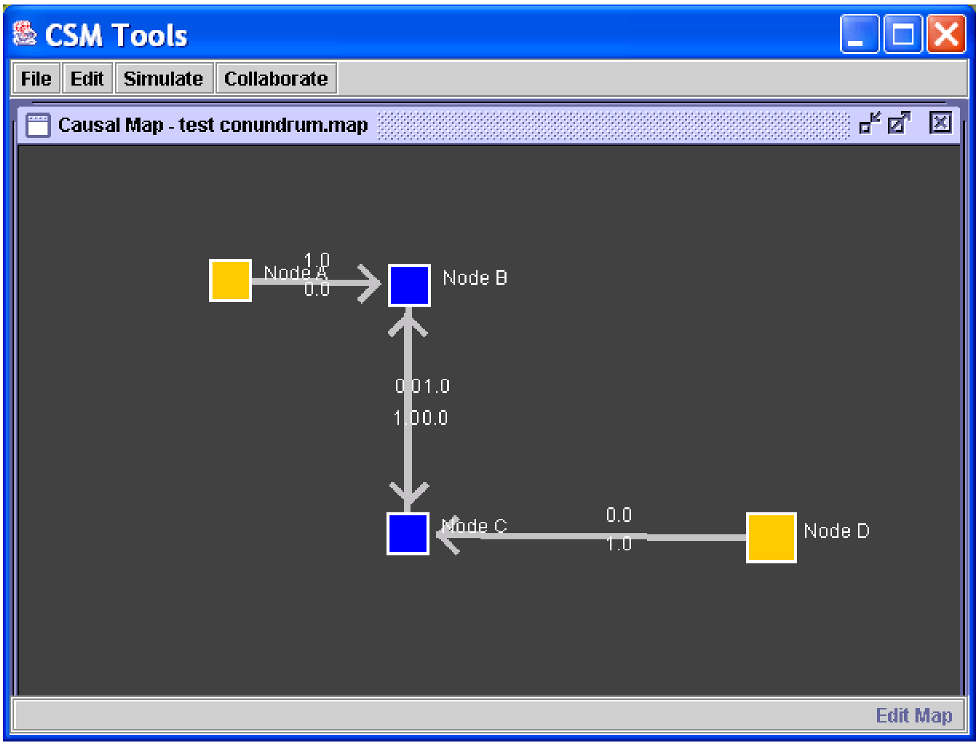


Figure 11. Test map of infinite causal loop.

Agents (nodes) are processed in order from A to D. Assume that, at time 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. This 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 but has not yet had a chance to act, so agent C has received no change from D. But the change from A via B is present, and C processes the change and places a change on the return link to B.

Now it is Agent D's turn to be processed. A change is placed on the link to C, and the link is activated. If agent behaviors were only scheduled for one-time activation during a time step, all processing stops and the simulation time is advanced. However, processing would be incomplete according to the semantics of the map, because there still is an unprocessed change on the D-C and C-B links. 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.

Avoiding Infinite Loops

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. 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 problem of infinite causal loop.

Two basic approaches can be taken to solving this problem. One approach is to limit the semantics so that time lags of zero 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 because a cause should play out prior to its effect (unless one is considering quantum mechanics) 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. In the current development we have chosen the former solution while working on alternatives along the latter mode.

8. Design Evaluation: Usability Testing Results

We conducted usability testing of the initial prototype according to standard procedures described in the usability testing literature (Rubin, 1994; Caulton, 2001; Zimmerman and Muraski, 1995). The details of these tests have been reported elsewhere in the literature (Druckenmiller et al., 2007), and so we will only summarize the findings here.

The interface must be intuitive and easy to use for widespread adoption. Graphic representation of dynamic situations should lead to new insight and the formulation of useful strategies. We conducted usability testing of the system to refine the user interface; it is an important aspect in the successful outcome of the use of a GSS in decision-making situations (Wheeler and Valacich, 1996). User acceptance and adoption is also dependent on the overall design of the strategic decision making process of which software is only a part. However, this testing is beyond the scope of the current development.

The design of a usability test is contingent on its exact purpose. Our exact purpose was to explore the basic usability of the current prototype compared with the original paper-based system. The testing focused on three aspects:

- first, the user interface issues in terms of its effectiveness, learnability, and likeability;
- second, a criterion for establishing baseline data for the completion of common tasks;
- and, third, a compilation of anticipated tasks, used to generate suggestions for product improvement.

The prototype testing results confirmed that the software tool was easy to learn and use, that it was preferred over the manual, paper-based system, and that several flaws in the user interface could be identified and improved. Further testing revealed that users prefer to use this tool for construction of causal maps over other generic drawing tools such as white boards, flow-charting tools, etc. Some CSM requisites such as the ability to reconfigure the maps and links dynamically in the tool would be extremely difficult or impossible to do with generic drawing toolsets. An even stronger case could be made in favor of our prototype's integrated notation system for minor assumptions (time lags and change-transfer coefficients), and its capability for carrying out the computation of the successive waves of changes needed to perform CSM's forward analysis of the potential scenarios implied by the causal graphs.

9. Limitations and Future Research Directions

The current prototype development has only considered the implementation of software tool support for creating, editing, and simulating causal maps' implicit scenarios. One of the key insights from Collaboration Engineering and the development of ThinkLets (Briggs et al., 2003) is the necessary specification of the facilitation script for use with a specific tool and its configuration. Further usability testing of facilitation scripts and associated requirements for *multiple* configurations will need to be done. The usability testing of the prototype compared it to a manual paper-based consulting method, and established the benefits of automation for ease of use and reduction of cognitive load, but further testing will need to be done to determine the facilitation approaches for creation and development of the kind of causal map that works best with the tool. Comparative testing with other causal mapping approaches and systems dynamics modeling is beyond the scope of the present development.

The iterative nature of software development, central to design science research and user-centered design, will necessitate successive developments as we learn more about how the tool affects group dynamics and the interaction between script and tool. Currently, we are using the tool with undergraduates in an Information Systems Management course to develop prototype facilitation models for strategic enterprise systems analysis and strategic project formulation. Our current tool has allowed students to make sophisticated and insightful analyses of complex strategic situations, but it will require further development for use in consulting situations involving top management levels.

The research described in this paper has many potential future directions in the development of additional software tools for systems thinking and business intelligence, as well as the development of scripting elements and new ThinkLets for the organizing pattern of interaction. CSM provides a framework for development of additional testing of the theory of problem formulation. Following are a few of the additional directions research could take:

- **Support for Backward Analysis**: In future developments, environmental scanners could be designed to use causal maps for automated intelligent filtering that provides a contextual framework for intelligent, directed environmental scans of the corporate external and internal environments. Business intelligence applications currently perform this role but lack strategic direction. One of the newest business intelligence 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 applications over the next decade (McCoy and Govekar, 2002; Gassman, 2003).
- **Support for Forward Analysis**: To gain competitive advantage through scenario-driven planning, strategic managers and their analytical routines need be designed to support "forward analysis" by using the simulation capabilities of causal maps such as those developed in the current prototype. A decade ago, Phelan (1997) used multi-agent systems in which he employed a learner-classifier (Holland, 1992) machine-learning simulation based on genetic algorithms. The central focus of Phelan's research was a simulation method using an adaptive multi-agent system that employed agents to simulate strategic situations that would be difficult to study experimentally (as is the case with strategy.)

  Phelan used agent systems to simulate the strategic landscape and the strategies pursued to optimize performance in a dynamic environment. While his research showed the benefits to be gained from creating a simulation test bed for testing generic strategies, it was limited to a micro-world testing of strategic theory instead of addressing how to model specific organizational situations. On the other hand, a multi-agent causal map could be used to model a specific strategic landscape, which could then beused to test development of robust adaptive strategies for specific organizational challenges.

- **Support for Structural Analysis:** Mapping facilitators need to be designed to support dialogue management for *graph analysis*, integration, and occasional comparison of causal maps. The key development here is the building of computerized agents to compile and analyze the factor or scope overlap of individual maps, and to propose potential changes to facilitate merging individual maps into a common one. Over time, agents could store graphs and use their change histories to refine the facilitation process. The pattern of changes and the recurring themes in graphs could be used by agents to help critique maps developed by end-users, and provide for an intelligent facilitation of the dialectical process of map creation.

10. Conclusion

This research develops an approach that theoretically stands between qualitative cognitive modeling and quantitative OR/MS modeling approaches. Following the design-science paradigm, and taking heed from the evolution of problem framing and structuring methods, we rigorously develop a specific conceptual model and an integrated tool for causal modeling. Our proposed tool addresses the key problem of the cognitive load introduced by quantitative System Dynamics modeling. The primary contribution of this development is an instantiation of this tool in a testable prototype system. The tool was evaluated through standard usability analysis for its ease of use and learnability.

Methodologically, we argue that there is a strong relationship between *Design Science* and *User-Centered Design*. User Centered Design and, specifically, usability testing provide a structured way to insure that rigorous theory-based design is also highly relevant. Thus, there is a natural partnership between design science research and user-centered design. This prototype illustrates how the two are complementary methodological thrusts that guide the development of software applications both theory-based and practical.

Strategically, this research contributes an operational linkage between scenario-driven planning for strategic management and the latest distributed artificial intelligence (DAI) modeling tools. Scenario-driven planning is revolutionizing the way decision makers manage uncertainty and change (Gassman, 2003; Rangarajan and Rohrbaugh, 2003; van der Heijden, 1996). Our development of a user-friendly distributed tool for creating and simulating causal maps demonstrates the synergy of well-designed methods for strategic thinking combined with intelligent support from distributed artificial intelligence features. Until this development, computerized modeling and simulation was deemed too difficult for the ordinary user and left to specialists (Eden and Ackermann, 1998; Ackermann and Eden, 2005). Our contribution paves the way for new tools combining a user-friendly interface for graphically developing scenarios and simulation models. This will put the design of the simulation model squarely in the hands of the user of the system and set the stage for the collaborative development of scenarios.

From now on, decision makers themselves will be able to develop and exploit the power of agent-based modeling and simulation at minimal cost and with maximal effectiveness. The research also opens the door to someday using agent-based conceptual models to guide intelligent searches of intranet and extranet space *to support the 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 is headed toward becoming an invaluable guide to selecting relevant information for strategic decision making.

References

Acar, M. A. (2000). *A Computerized, Collaborative System for the Creation and Edition of Causal Maps.* B.S. Thesis, Kent, Ohio: 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.

Acar, W. and D. A. Druckenmiller (2006). "Endowing Cognitive Mapping With Computational Properties for Strategic Analysis", *Futures,* (38) pp.993-1009.

Ackermann, F. and C. Eden (2005). *The Practice of Making Strategy: A Step by Step Guide*. London: Sage.

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

Andersen, D. F., G. P. Richardson, and J. A. M. Vennix (1997). "Group Model Building: Adding More Science to the Craft", *System Dynamics Review,* (13)2, pp.187-201.

Andersen, D. F., J. M. Bryson, G. P. Richardson et al. (2006). "Integrating Modes of Systems Thinking into Strategic Planning Education and Practice: The Thinking Person's Institute Approach", *Journal of Public Affairs Education* (12)3, pp. 265-293.

Berente, N. and K. Lyytinen (2006). "The Iterating Artifact As a Fundamental Construct in Information System Design", In *Proceedings DESRIST,* Claremont, CA: Claremont Graduate University. http://ncl.cgu.edu/designconference/DESRIST%202006%20Proceedings/6B_3.pdf

Briggs, R. O., G. J. de Vreed, and J. F. Nunamaker (2003). "Collaboration Engineering With ThinkLets to Pursue Sustained Success With Group Support Systems", *Journal of Management Information Systems,* (19)4, pp.31-64.

Bryson, J. M., F. Ackermann, C. Eden et al. (2004). *Visible Thinking: Unlocking Causal Mapping for Practical Business Results*. Wiley.

Carlsson, S. A. (2006). "Towards an Information Systems Design Research Framework: A Critical Realist Perspective", In *Proceedings DESRIST 2006,* Claremont, CA: Claremont Graduate University. http://ncl.cgu.edu/designconference/DESRIST%202006%20Proceedings/6B_1.pdf

Caulton, D. A. (2001). "Relaxing the Homogeneity Assumption in Usability Testing.", *Behavior and Information Technology,* (20)1, pp.1-7.

Chaib-draa, B. (2002). "Causal Maps: Theory, Implementation, and Practical Applications in Multiagent Environments", *IEEE Transactions on Knowledge and Data Engineering,* (14)6, pp.1201-1217.

Checkland, P. (1981). *Systems Thinking, Systems Practice*. London: Wiley.

Churchman, C. W. (1968). *The Systems Approach*. New York: Dell.

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

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

Dennis, A. R. and M. J. Garfield (2003). "The Adoption and Use of GSS in Project Teams: Toward More Participative Processes and Outcomes", *MIS Quarterly,* (27)2, pp.289-323.

Dennis, A. R., C. Tyran, D. R. Vogel et al. (1997). "Group Support Systems for Strategic Planning", *Journal of Management Information Systems,* (14)1, pp.155-184.

Druckenmiller, Douglas A., W. Acar et al. (2007). "Usability Testing of an Agent-Based Modelling Tool for Comprehensive Situation Mapping", *International Journal of Technology Intelligence and Planning* (3)2, pp. 193-212.

Eden, C. (1990). Using cognitive mapping for strategic options development and analysis – SODA. In J.Rosenhead (Ed.), *Rational Analysis for a Problematic World* (.

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

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

Franco, L. A. (2006). "Forms of Conversation and Problem Structuring Methods: A Conceptual Development", *Journal of the Operational Research Society* (57), pp. 813-821.

Fulk, J. and G. DeSanctis (1995). "Electronic Communication and Changing Organizational Forms", *Organization Science,* (6)4, pp.337-349.

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

Gartner Research.

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

Gregor, S. and D. Jones (2007). "The Anatomy of a Design Theory", *Journal of the Association for Information Systems (JAIS),* (8)5, p.Article 19.

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

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

Hender, J. M., D. L. Dean, T. L. Rodgers et al. (2002). "An Examination of the Impact of Stimuli Type and GSS Structure on Creativity: Brainstorming Versus Non-Brainstorming Techniques in a GSS Environment", *Journal of Management Information Systems,* (18)4, pp.59-85.

Hevner, A. R., S. T. March, J. Park et al. (2004). "Design Science in Information Systerms Research", *MIS Quarterly* (28)1.

Hodgkinson, G. P., N. J. Bown, A. J. Maule et al. (1999). "Breaking the Frame: an Analysis of Strategic Cognition and Decision Making Under Uncertainty", *Strategic Managemetn Journal,* (20) pp.977-985.

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

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

Huff, A. S. and M. Jenkins (2002). *Mapping Strategic Knowledge*. London: SAGE Publications Ltd.

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

Kettelle, J. (2006). "When Three's Not a Crowd", *OR/MS Today,* (33)5, pp.20-27.

Knudsen, T. L. D. A. (2007). "Two Faces of Search: Alternative Generation and Alternative Evaluation", *Organization Science* (18)1, pp. 39-54.

Kuhn, T. (1996). *The Structure of Scientific Revolutions*. Chicago: University of Chicago Press.

Maruyama, M. (1963). "The Second Cybernetics: Deviation-Amplifying Mutual Causal Processes", *American Scientist,* (51) pp.164-179 & 250-256.

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

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

Montibeller, G. and V. Belton. (2006). "Causal Maps and the Evaluation of Decision Options - a Review", *Journal of the Operational Research Society* (57), pp. 779-791.

Panepento, Peter (2000). "The Perfect Swarm", *Computerworld*.

Peachey, T. A. and D. J. Hall. (2006). "Supporting Complex Problems: an Examination of Churchman's Inquirers As a Knowledge Management Foundation", *Knowledge Management Research & Practice* (4), pp. 197-206.

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

Potter, R. E. and P. Balthazard (2004). "The Role of Individual Memory and Attention Processes During Electronic Brainstorming", *MIS Quarterly,* (28)4, pp.621-643.

Rangarajan, N. and J. Rohrbaugh (2003). "Multiple Roles of Online Facilitation: An Example in Any-Time, Any-Place Meetings", *Group Facilitation,* (5) p.26.

Richmond, B. (1994). "System Dynamics/Systems Thinking: Let's Just Get on With It", In Sterling Scotland: International Systems Dynamics Conference.

Robertson, D. A. (2003). "Agent-Based Models of a Banking Network As an Example of a Turbulent Environment: the Deliberate Vs. Emergent Strategy Debate Revisited", *Emergence,* (5)2, pp.56-71.

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

Scavarda, A. J., T. Bouzdine-Chameeva, S. Meyerr Goldstein et al. (2006). "A Methodology for Constructing Collective Causal Maps", *Decision Sciences* (37)2, pp. 263-283.

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

Simon, H. (1996). *The Sciences of the Artificial*. (Third Edition ed.) Cambridge, MA: MIT Press.

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

Sun Microsystems, I. J. (1999). *Java Look & Feel Design Guidelines, 1st Edition*. Boston, MA: Addison-Wesley Longman Publishing Co., Inc.

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

Tyran, C. K., A. R. Dennis, D. R. Vogel et al. (1992). "The Application of Electronic Meeting Technology to Support Strategic Management", *MIS Quarterly,* (16)3, pp.313-334.

University of Chicago: Social Science Research Computing (2002). "RePast 2.0", [Computer software]. http://repast.sourceforge.net/.

Vahidov, R. (2006). "Design Researcher's IS Artifact: a Representational Framework", In *Proceedings DESRIST 2006,* Claremont, CA: Claremont Graduate University. http://ncl.cgu.edu/designconference/DESRIST%202006%20Proceedings/2A_2.pdf

van der Heijden, K. (1996). *Scenarios: The Art of Strategic Conversation*. Chichester: Wiley.

Vennix, J. A. M., D. F. Andersen, and G. P. Richardson (1997). "Foreword: Group Model Building, Art, and Science", *System Dynamics Review,* (13)2, pp.103-106.

Wheeler, B. C. and J. S. Valacich (1996). "Facilitation, GSS, and Training As Sources of Process Restrictiveness and Guidance for Structured Group Decision Making: An Empirical Assessment", *Information Systems Research,* (7)4, pp.429-450.

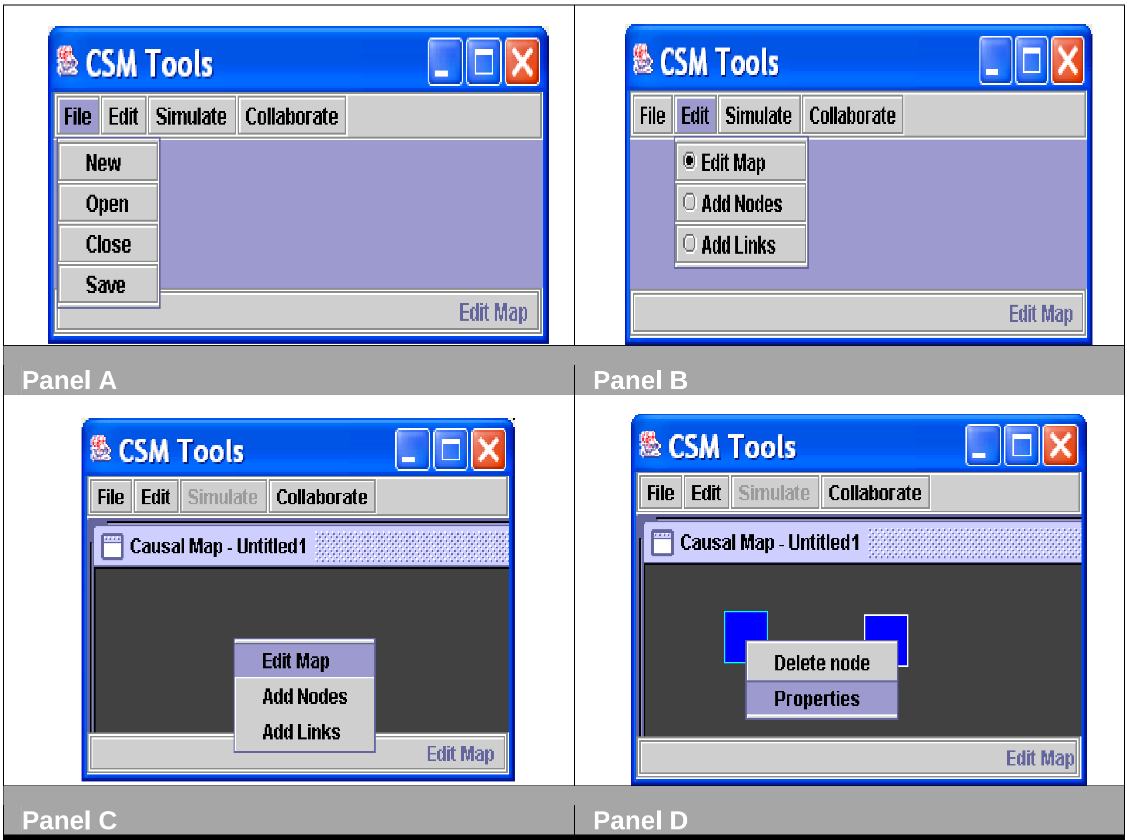
White, L. (2006). "Evaluating Problem-Structuring Methods: Developing an Approach to Show the Value and Effectiveness of PSMs", *Journal of the Operational Research Society* (57), pp. 842-855.

Zhiang, J. L., X. Zhao, K. M. Ismail et al. (2006). "Organizational Design and Restructuring in Response to Crises: Lessons From Computational Modeling and Real-World Cases", *Organization Science,* (17)5, pp.598-618.

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

Appendix A: CSMTools user interface

Our current user interface components (see Figure 12) are based on drop-down menus that follow common user-interface 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 and *save* the current map (see panel A).



Panel A

Panel B

Panel C

Panel D

Figure 12. User interface components

Basics of the User Interface

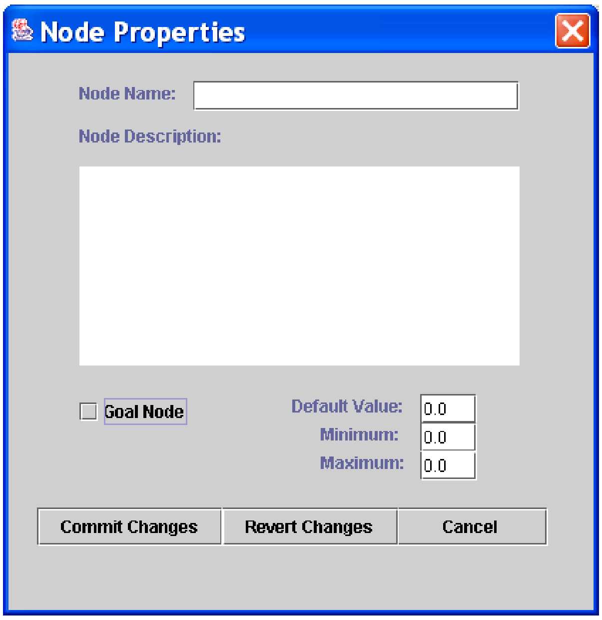
The **"Edit" menu** allows the user to edit the map, add nodes and add links (see Figure 12, panel B). In the "Edit Map" mode, the user may select and reposition nodes and links on the map. When "Add Nodes" is selected, the user can add nodes with each left mouse click. When "Add Links" is selected, 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 Map" mode where nodes and links can be repositioned, deleted or edited, and two "Add" modes where nodes and links may be added to the drawing surface but not edited. This is an efficient way to draw maps because the user is not required to continually switch between the Add and Edit modes 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 not used to doing things this way.
Also, in the "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 that node (see Figure 12, panel D).

More Advanced Features

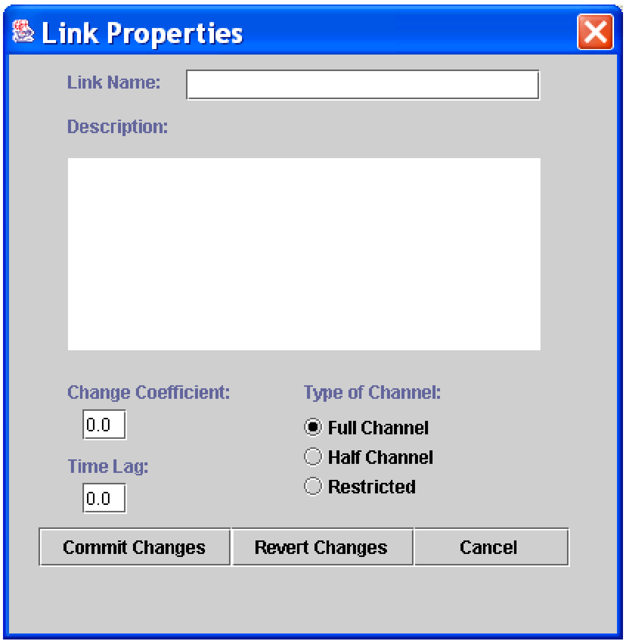
Choosing the **“Properties” option** allows the user to edit the node/link property sheets. “**Error! Reference source not found.**” shows the node property sheet. Fields are available to enter the node name, description, default value, minimum and maximum threshold values, and a check box that indicates whether the node is a goal node. If the goal node property is checked, its value is automatically graphed in a simulation allowing the user to evaluate goal expected values. Command buttons allow changes to be saved (“Commit Changes”) or discarded (“Revert Changes”); another option is to discarde changes and close the property sheet (“Cancel”).



Default Value: 0.0
Minimum: 0.0
Maximum: 0.0

Figure 13. Node property sheet.

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-transfer coefficient, time lag and type of channel. Buttons allow changes to be saved (Commit Changes), discarded (Revert Changes), or again discarding changes and closing the property sheet (Cancel).



Link Properties

Link Name:

Description:

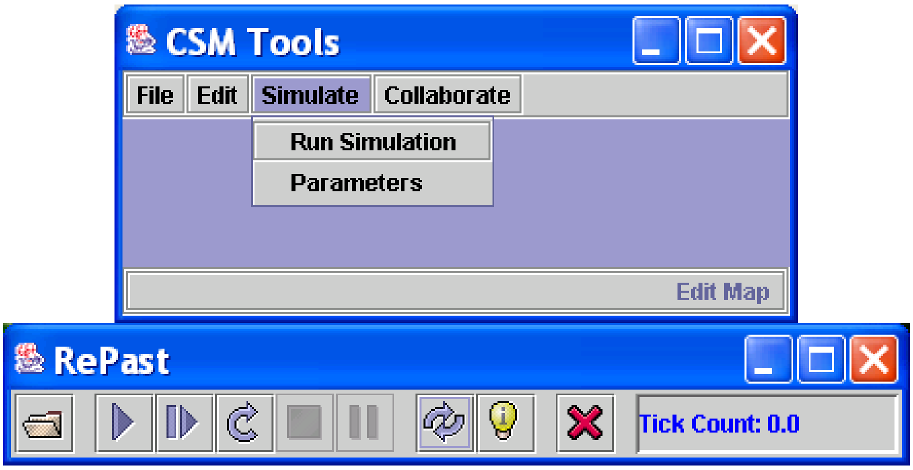
Change Coefficient: 0.0

Time Lag: 0.0

Type of Channel:
● Full Channel
○ Half Channel
○ Restricted

Figure 14. Link property sheet.

The **"Simulate" menu** allows you to simulate the map with the "Run Simulation" item, even though the "Parameters" menu item may not be not currently set-up. When the "Run Simulation" item is chosen the simulation tool bar is displayed (see Figure 15).



Tick Count: 0.0

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 <u>next</u> 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 re-started.

✖ The cancel button, which exits the simulation function.

About the authors

Douglas A. Druckenmiller is an Assistant Professor in the Department of Information Systems and Decision Sciences at Western Illinois University. Dr. Druckenmiller received his PhD in Management Information Systems from Kent State University. His research focuses on design research in collaboration engineering; development and testing of software to support facilitation processes involving causal mapping, agent-based modeling and simulation tools, systems dynamics modeling and thinking, Group Support Systems (GSS) and virtual meeting technologies. He is currently principle investigator of a major international dual-degree project in business and technology involving multiple United States and European Union partner universities and in virtual communities of practice. He has more than 35 years of collaboration engineering experience, and has published several journal articles in the area of group decision support and problem formulation in journals such as *The International Journal of Intellegence, Technology and Planning*, *FUTURES,* and *Journal of Information Systems Education.*

William Acar (Dipl. Ing.; M.A.Sc. Waterloo, CA; Ph.D. Wharton-Upenn, USA) teaches Management and Information Systems at Kent State University. The author of numerous published articles, Dr. Acar is the initial author of the CSM causal mapping method for the analysis of complex business situations and coauthored a book entitled ***Scenario-Driven Planning*** that referred to it. Currently, his method is being used to develop a computerized GSS for solving strategic and organizational learning problems. He has also developed measures of diversity better calibrated than the Herfindahl index and the Entropy function. He has published in journals such as: *Strategic Management Journal, Journal of Management Studies, INFOR, Decision Sciences, OMEGA, Journal of Management, Information Systems*, *European Journal of Operational Research, International Journal of Operational Research, Systems Research, Behavioral Science*, *FUTURES, International Journal of Organizational Analysis, International Journal of Technology Management, Production & Inventory Management, Strategic Change, INTERFACES* and the *International Journal of Commerce & Management.*

Journal of the Association for Information Systems

ISSN: 1536-9323

Editor
Kalle Lyytinen
Case Western Reserve University, USA

Senior Editors			
Robert Fichman	Boston College, USA	**Dennis Galletta**	University of Pittsburgh, USA
Varun Grover	Clemson University, USA	**Rudy Hirschheim**	Louisiana State University, USA
Robert Kauffman	University of Minnesota, USA	**Frank Land**	London School of Economics, UK
Jeffrey Parsons	Memorial University of Newfoundland, Canada	**Suzanne Rivard**	Ecole des Hautes Etudes Commerciales, Canada
Ananth Srinivasan	University of Auckland, New Zealand	**Bernard C.Y. Tan**	National University of Singapore, Singapore
Michael Wade	York University, Canada	**Ping Zhang**	Syracuse University, USA
Editorial Board			
Steve Alter	University of San Francisco, USA	**Kemal Altinkemer**	Purdue University, USA
Michael Barrett	University of Cambridge, UK	**Cynthia Beath**	University of Texas at Austin, USA
Michel Benaroch	University of Syracuse, USA	**Francois Bodart**	University of Namur, Belgium
Marie-Claude Boudreau	University of Georgia, USA	**Susan A. Brown**	University of Arizona, USA
Tung Bui	University of Hawaii, USA	**Andrew Burton-Jones**	University of British Columbia, Canada
Dave Chatterjee	University of Georgia, USA	**Patrick Y.K. Chau**	University of Hong Kong, China
Mike Chiasson	Lancaster University, UK	**Mary J. Culnan**	Bentley College, USA
Jan Damsgaard	Copenhagen Business School, Denmark	**Samer Faraj**	McGill university, Canada
Chris Forman	Carnegie Mellon University, USA	**Ola Henfridsson**	Viktoria Institute & Halmstad University , Sweden
Hitotora Higashikuni	Tokyo University of Science, Japan	**Kai Lung Hui**	National University of Singapore, Singapore
Hemant Jain	University of Wisconsin-Milwaukee, USA	**Bill Kettinger**	University of South Carolina, USA
Rajiv Kohli	College of William and Mary, USA	**Mary Lacity**	University of Missouri-St. Louis, USA
Ho Geun Lee	Yonsei University, Korea	**Jae-Nam Lee**	Korea University
Kai H. Lim	City University of Hong Kong, Hong Kong	**Ji-Ye Mao**	Renmin University, China
Anne Massey	Indiana University, USA	**Emmanuel Monod**	Dauphine University, France
Michael Myers	University of Auckland, New Zealand	**Fiona Fui-Hoon Nah**	University of Nebraska-Lincoln, USA
Mike Newman	University of Manchester, UK	**Jonathan Palmer**	College of William and Mary, USA
Paul Palou	University of California, Riverside, USA	**Brian Pentland**	Michigan State University, USA
Yves Pigneur	HEC, Lausanne, Switzerland	**Jaana Porra**	University of Houston, USA
Sandeep Purao	Penn State University, USA	**T. S. Raghu**	Arizona State University, USA
Dewan Rajiv	University of Rochester, USA	**Balasubramaniam Ramesh**	Georgia State University, USA
Timo Saarinen	Helsinki School of Economics, Finland	**Susan Scott**	The London School of Economics and Political Science, UK
Ben Shao	Arizona State University,USA	**Olivia Sheng**	University of Utah, USA
Carsten Sorensen	The London School of Economics and Political Science, UK	**Katherine Stewart**	University of Maryland, USA
Mani Subramani	University of Minnesota, USA	**Burt Swanson**	University of California at Los Angeles, USA
Dov Te'eni	Tel Aviv University, Israel	**Jason Thatcher**	Clemson University, USA
Ron Thompson	Wake Forest University, USA	**Christian Wagner**	City University of Hong Kong, Hong Kong
Eric Walden	Texas Tech University, USA	**Eric Wang**	National Central University, Taiwan
Jonathan Wareham	ESADE, Spain	**Stephanie Watts**	Boston University, USA
Bruce Weber	London Business School, UK	**Tim Weitzel**	Bamberg University, Germany
Richard Welke	Georgia State University, USA	**George Westerman**	Massachusetts Institute of Technology, USA
Kevin Zhu	University of California at Irvine, USA	**Ilze Zigurs**	University of Nebraska at Omaha, USA
Administrator			
Eph McLean	AIS, Executive Director	Georgia State University, USA	
J. Peter Tinsley	Deputy Executive Director	Association for Information Systems, USA	
Reagan Ramsower	Publisher	Baylor University	