Engineering Dialectical Inquiry: Lessons Learned from Lab Explorations

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

A Design Science approach is applied to the engineering of a dialectical inquiry process for group model building. Insights from three research streams in collaboration literature are used for the practical engineering of a facilitation script. Dialectical inquiry is the process developed by Churchman and used in Comprehensive Situation Mapping (CSM) as recently described by Acar and Druckenmiller. The software artifact developed for this approach was tested with usability testing to compare the ease of use of this computerized technique with the original manual method. While this software was easy to learn and to use, its development did not fully address the accompanying facilitation model that provided a tested script for use in actual field cases. This research uses the principles of user-centered design as an iterative search component of the Design Science approach. Its application to the development of such scripts, as well as the initial results of an exploratory phase of testing are described.

1. Introduction

The great French diplomat Talleyrand once said that “the art of statesmanship is to foresee the inevitable and to expedite its occurrence.” Foreseeing the inevitable is the task of organizational foresight. Expediting the inevitable allows that which is foreseen to become current reality. The process of foreseeing the inevitable is a complex process that brings the insight and knowledge of multiple individuals within an organization into a single vision of the future that is both inevitable and implementable. Modern organizational foresight seeks to make this an engineered process not solely dependent on great statesmen or the talents of unique individuals, but becoming part of an ongoing sustainable and repeatable process. Current research in organizational foresight has focused more on the unique creations of individual researchers who pioneered foresight systems. Little attention has been paid to the issue of how foresight systems become a sustainable and repeatable process that enables continuous use of such systems within organizations.

2. Research Methodology

The design science paradigm [12; 26; 29; 45] is used as The methodological approach. 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 [6; 31; 41]. Hevner et al. [29] provide guidelines for design science research. Design Science research is both rigorous in that it is theory based and relevant in that it uses an iterative search approach to designing artifacts that test the theory.
This paper first draws from relevant theory for the practical engineering of a facilitation model for dialectical inquiry. Dialectical inquiry (DI) is the interactive process adapted from Churchman [13] and used in the development of causal maps in Comprehensive Situation Mapping (CSM) as described by Acar and Druckenmiller [2]. The software developed for this approach [20] was tested for usability in a subsequent study[19], to compare the ease of use of our software product with the original CSM manual method. While the software was easy to use and learn from the start, its initial development did not address the accompanying facilitation model that provided a tested script for use in actual field cases and studies.

The paper then describes an exploratory search process using user-centered design techniques to iteratively develop a process for dialectical inquiry. This process is expressed as a pattern language artifact consistent with collaboration engineering principles [7]. Lessons learned from the results of the exploratory phase of testing are reported and the next phase of the research is briefly discussed in the concluding section.

3. Theory Development

Three research streams in the field of collaborative systems: i) conceptual mapping and group model building, ii) systems thinking and group model building, and iii) collaboration engineering. Insights from these three research streams are used for the practical engineering of a facilitation model for dialectical inquiry.

3.1. Conceptual Mapping and Group Model Building

Cognitive approaches to organizational foresight have focused on the challenge of finding analytical approaches to the synthetic and creative process of strategy conceptualization. Intuition can be informed and creative processes can be structured. This approach integrates the emergent strategy notion of the incrementalists (e.g., Mintzberg et al., [34] with the deliberately rational approach advocated by the proponents of planning (e.g., Ansoff; [4; 5]). Cognition is important in confirming and redesigning strategy, for managing complexity and organizational change, for strategic decision making and problem solving, and for enabling cognitive change and flexibility [22]. Eden’s approach to strategy making uses a continuous cycle of learning through a review of both emergent and planned strategies. The process is an integration of a variety of schools of strategy and is described as a “journey of strategy making” [22].

Broadly speaking, cognitive mapping refers to several modes of graphic representation of individual perceptions and beliefs about a situation. Graphic representations are well known in the management and social science literature as both systems analysis tools and knowledge representation techniques [42]. A number of specific approaches have been developed, including adaptations of mathematical graph theory [23], cognitive psychology and personal construct theory [30], influence diagramming [32] and system dynamics [23].

In their book Making Strategy, Eden and Ackermann [22] detail the process for developing conceptual maps as a group model building process that elicits the structure of strategic problems. Their group facilitation process has become a script for use with a software tool that results in the consistent development of usable conceptual maps for strategic analysis.

At the same time researchers in systems thinking were focused on the importance of the facilitation script for group model building. Andersen & Richardson [3] acknowledge the foundational work of Eden and Ackermann and conceptual mapping in their approach to group model building (GMB). GMB has been taken up by the “systems thinking” movement in System Dynamics.

3.2. Systems Thinking and Group Model Building

These approaches attack the problem with group dynamics modeling by users within a systematic framework of scripted activities. These “scripts” are specific group activities that guide the thinking process and produce the results of a consultative intervention. The various scripted components are woven together in larger patterns of interaction which together produce the results of a consultation. Systems thinking researchers have developed and documented various useful scripts over the past 20 years our their practical consulting experience. These scripts can provide a basis for the specific development of specific thinking components for use with CSM. Systems thinking scripts are developed for:

- defining the problem,
- conceptualizing model structure,
- eliciting feedback structure,
- equation writing and parameterization,
- policy development (but not testing).

However, assessing the affects of group model building interventions has not been systematic.
Research is mostly based on case studies; rigorous approaches to scientific testing and validation of the effectiveness of the process is needed [39].

Barry Richmond [38] and his company, High-Performance Systems (HPS), developed the user-friendly software STELLA as a way to make System Dynamics modeling more accessible to the ordinary user and has been quite successful in increasing the adoption of System Dynamics. According to Richmond three impediments exist to representing complex mental models as a series of stocks and flows:

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

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. A second refinement is a “space-compression object” that simplifies the complexity of the map and provides an easier way to model causal loop formations. This is precisely the insight that is at the foundation of CSM. Richmond says that:

…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 [38], p. 12).

3.3. Collaboration Engineering

Meanwhile a similar journey has been taking place in the GSS community. The majority of GSS research has been in electronic brainstorming or ideation. Creative approaches to ideation and the role of ideation in deeper thinking processes is little explored however. Hender et al. [28] report that “Of the techniques available for idea generation with group support systems (GSS), little research attention has been given to techniques that challenge problem assumptions or that use unrelated stimuli to promote creativity”. This research used assumption reversals for improving the number of creative ideas in group brainstorms but general produced fewer creative ideas. Other research [36] identifies the benefits of “cause cueing” which directs attention to the underlying causes of problems identified by participants. Current tools work well for generation of large numbers of ideas through electronic brainstorming but generating consensus and converging on key decisions remains problematic especially in conflict-resolution situations [37].

The potential for GSS systems support of strategic management communication processes and qualitative analysis has been the subject of research for many years [44]. Specific communication technologies have an interdependent relationship with organizational form. The interplay between form and technology is an important aspect of organizational change, and the integration of computer technology and communications capabilities has meant a richer communication capability for organizations [24]. In a study involving 30 organizations who used GSS for strategic planning activities, Dennis, Tyran, Vogel & Nunamaker [18] found that “the ability of the GSS to provide process support was found to be the most important contributor to strategic planning success”.

Creative research projects have combined business-process reengineering (BPR) with electronic meeting systems [16] creating Collaborative Business Engineering. GSS collaborative mechanisms using simulation [15] and process modeling [14] are used to overcome insufficient stakeholder involvement, poor analysis of business processes, and the quality of modeling in BPR projects leading to improvements in BPR project success.

Technology supported facilitation is also increasingly important in the facilitation of virtual teams. Virtual teams are an important aspect of strategic flexibility of today’s firms and are increasingly dependent on networked communications technology for their interaction. A richer form of communication is needed however to offset communication losses from the restricted bandwidth currently in use [43]. GSS-based global virtual teams and the diversity of individualistic/collaborative style on group performance and effectiveness is a key research area. The diversity of group members enhanced decision quality when combined with collaborative conflict resolution style and GSS [35]. Other researchers report on the satisfaction with GSS outcomes and consensus as primarily influenced by cultural aspects of the participants [33].

In spite of its successes the research in GSS has been inconsistent. Some research validates the usefulness of GSS in organizational decision making while other studies are not as glowing [10]. Like systems dynamics, GSS has seen great promise but long term use and adoption by organizations has been slow. Despite initial success long term use is rare. Adoption and diffusion research has focused on the issues of slow adoption of GSS in organizations and project teams and factors influencing sustained use of GSS [9; 17].
Central to understanding the reasons for success or failure is the Technology Transfer model developed by Briggs, de Vreede and Nunamaker [8]. Researchers found that, while the perceived ease of use was high when professional facilitators configured systems and led group processes, the same systems were confusing and fell into disuse when facilitation was not available. Foresight is a complex social and conceptual process that requires an expert facilitation of group dynamics and integration with foresight methods. Successful implementation of self-sustaining GSS systems was found to depend on two factors: i) specific processes and technology were engineered to create solutions for mission-critical tasks that recur frequently; ii) skilled facilitators were used to design specific effective collaboration processes and train skilled practitioners to run them.

The key to the successful implementation of complex organizational foresight mechanisms is the design of specific processes and the training of practitioners and groups that will become self-sustained users of the process. This approach is termed collaboration engineering. It requires: a) facilitations skills; b) the development of step-by-step processes; c) standard packages to become reusable building blocks that can easily be recombined for supporting other processes.

Collaboration engineering identifies these standard packages as “ThinkLets” [7]. ThinkLets are a technology independent pattern language for describing facilitation processes. In particular ThinkLets describe standard facilitation scripts for group processes. The facilitation script is an important aspect in the successful outcome of the use of a GSS in decision making situations [46]. ThinkLets represent an attempt to minimizing cognitive load for practitioners an important impediment to sustained use of GSS in organizational settings.

3.4. CSM and a Dialectical Inquiry Approach to Group Model Building

Our approach to implementing collaboration engineering is Comprehensive Situation Mapping (CSM) [1; 2; 20; 25]. CSM is a causal mapping approach combined with a process of dialectical inquiry. It is specifically configured to model the way group members visualize situations and to capture the pattern of their dialectical interactions. The tool for these configurations is the CSM software developed by the authors [20] for stand-alone or Internet applications. It provides a means for successfully imbedding dialectical inquiry as a pattern of interaction in organizations as scripted by the CSM approach.

4. Iterative Search Process

Design Science Research uses an iterative search process to test theory through the development of specific artifacts that provide empirical evidence for the validity of the theory. Usability testing is a stand approach for an iterative design search process and is well suited for use in Design Science Research.

4.1. The Usability Testing Approach

Usability testing provides feedback on the prototype that can be used to improve the system. The purpose of usability testing is to improve the system’s ease of use, quality, user satisfaction and user time on task. The usual issues of usability testing and their attendant problems comprise subject sampling (pointing to the generalizability of the results), test implementation (experimental design of the tests), validity and reliability of the testing (evaluation and coding of test tasks), and interpretation of the results (problems of experimental bias and subjective evaluation) [47].

The data collected for evaluation purposes usually benefit from “triangulation” entailing a multi-method approach intended to minimize errors and maximize resulting insights. Protocol analysis is most typically used. Namely, as typical tasks are defined and are given to subjects to complete, all subjects talk out loud while executing the tasks and the test is videotaped. The evaluators can then review the test, analyze the data and interpret the results. These are occasionally complemented by direct observation as well as personal interviews and surveys. In all cases, clarity in identifying the variables and pertinent coding schemes is a requisite of proper evaluation and interpretation of the data.

4.2. User-Centered Design and its Goals

Usability testing is part of an emphasis on User-Centered Design (UCD) [40)). UCD is a process that designs the relationship between software or hardware products and their users with an emphasis on how the design enables user-centered tasks. This process is intimately connected to prototype development. The goal of UCD has been to fit the design of the system to the needs of the user, rather than fit the user to the needs of the system. Intriguingly, the former process is often found to be easy to learn and yet supports effective task accomplishment. Vexingly, the later is difficult to use and learn, requires expensive training, and yet often results in low user acceptance.
The principles of UCD emphasize an iterative development process because its initial design phase focuses mostly on the users and their tasks. It is only after that initial first step that subsequent iterations of empirical testing and corrective redesign can be undertaken. UCD is thus a continual cycle of prototype redesign, modification and testing. The emphasis on iterative UCD has shifted information technology from its past focus on technical implementation, which can now be automated or outsourced, to user interface design. Three components of functionality are hence integral to the development of usable systems: a) the user interface, b) the user-assistance system, and c) the documentation appertaining to these A and B components.

In line with this approach, the development of competitive products in UCD mode utilizes an emergent approach of experiential learning, yet one accomplished in a rational framework. Methods for UCD Implementation

A number of methods have been used for UCD, including participatory design, focus group research, surveys, design walk-throughs, paper and pencil evaluations, expert evaluations, usability audits, usability testing, field studies and follow-up studies. As alluded to earlier, usability studies are either formal “summative” studies conducted as true experiments with a hypothesis to be tested (not our current concern) or are “formative” assessments designed to expose usability deficiencies. The formative assessments of interest to us are of four types: exploratory, assessment-oriented, validation testing and comparative. Comparison studies are most often done at the exploratory stage but can be combined with each of the other three stages of testing. Combining the validation and comparison stages shortens the cycle to just three phases:

4.2.1 Exploratory testing of the prototype.

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

4.2.2 Assessment testing.

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

4.2.3 Verification testing.

Prior to final product release, verify that the system meets all functional requirements and make minor adjustments to the product before use by the end user. A full-release product requires 100% functionality, including a help system and documentation as mentioned above.

5. Exploratory Testing of Dialectical Inquiry Scripts

In order to develop a usable facilitation script for CSM a suitable test environment must be constructed. This represents a substantial problem for the designer, in that the dynamics of testing complicated group processes and procedures must be done in an environment where failure of the approach is always possible and the test user group is a proxy for an actual group of management decision makers. This type of testing is necessary before full field testing and development is possible.

5.1. Exploratory Approach

For this reason an appropriate class of upper division undergraduates was selected. The course focused on the strategic management of information systems. This course looks at how the competitive advantage of firms can be enhanced through the strategic use of information technology. The course uses a case study approach in which students are engaged in a process of researching and writing a business case that loosely follows a classic case study approach. Students work in small teams of three and spend a semester writing the case in a series of three assignments. There have been six iterations (two per year) of this process so far. The state of the usability framework evolved with each iteration, and refinements from the research findings of each iteration have been applied to the framework. Thus a systematic investigation of the use of CSM as both a teaching and analytical thinking tool has been conducted.

The first two iterations of testing focused on the development of the testing scenario which utilized a classic case study approach to provide a context and problem that required analysis. A classic case study approach can be accomplished in three stages. In the first stage a company of interested is selected and basic research into the facts of the case is conducted. Here students are asked to select a company of interest and review the published information about the company. If not enough written information can be found an alternative company must be selected. Students are
introduced to Michael Porter’s competitive forces and value chain models and these are used as an initial conceptual model to detail the facts of the case. In the second stage of the study, the students are asked to articulate a single strategic problem that is the root problem of the case. In the third stage of analysis the teams are asked to identify emerging information technologies. This also requires research into current information technology trends. Users are asked not to only identify trends that seem relevant to the company’s situation and problem but also to gain a broad overview of emerging information technology. Students then propose solutions to the case problem that involve innovative use of emerging information technologies.

This approach is limited due to the limited knowledge of students about the actual strategic situation. However, the purpose of the usability testing in this case is exploratory. Part of the purpose of this phase of testing is to establish basic procedures and scripts and to develop preliminary facilitation models that can be more thoroughly tested under field conditions. There is also a certain value in understanding the aspects of learnability with novice users.

In the next two iterations of the testing the focus was on developing the training that would be needed for students to understand the concepts of the causal mapping system, develop initial causal maps and enter them into the software. A training module was developed that utilized a published article on the pharmaceutical industry. The process detailed for developing a causal map by Acar [1] was followed. Its first phase is a survey phase where the actors and factors are listed. This is followed by a structuring phase in which the factors are linked into a causal map. A final phase details the type of linkages and adds simulation information in terms of magnitude of change and time lags. Students work together in the survey stage using a brainstorming technique and a group list of factors is developed. From a collaboration engineering perspective this is first a “divergence” pattern followed by a “convergence” pattern that creates an agreed list of factors. The next stage uses an organizing pattern of interaction which focuses on the relationships between the factors, and then elaborates those relationships with additional specification. This overall pattern is consistent with the systems thinking literature on first developing specific factors and linking them together into problem structure.

The final two iterations used the same case study approach but integrated causal mapping into each stage of the case as part of the analysis. Once the basic facts of the case are identified students are introduced to CSM concepts through the developed training session and are then asked to construct an initial situation map of the company’s competitive situation. The students then present their findings as both a written report and an oral presentation. This allows for an extended examination of the map contents and refinement of the basic elements and their relationships. Probing questions and opposing points of view are surfaced and discussed as the process of dialectical inquiry is used to refine the understanding of the competitive situation of the organization being studied. At this point an additional refining step is needed. Students typically put into a map their current conceptual view of the situation and since Porter’s categories on competitive forces have been used to identify the competitive facts in the case.

5.2. Lab result Details

The following map (Figure 1) is a typical first group modeled map and is illustrative of the problem. The factors in the map are mainly Porter’s competitive forces model with some other generic business factors added in. From a pedagogical standpoint, this reveals the students current level of perception about the case and the map is in need of refinement through dialectical inquiry.

Figure 1. Initial Student View

The primary question used at this point is to push for more specific information about each of the nodes on the map. Dialectical inquiry challenges the assumptions of each element of the map and pushes to surface hidden assumptions. The group is asked to come up with specific illustrations involving the company for each node and then the relevant actors and factors are identified. This resulted in the following refined model (Figure 2).
The next phase of the study is a determination of the central strategic problem that the organization faces and that is represented in the map. This entails an investigation of underlying or root causes and forward analysis of the map is introduced. This involves first of all a set of assumptions about the minor elements of the map linkages. Students must locate research that indicates time lags and coefficients of change. At the beginning the software defaults are accepted. All changes are proportional changes and the time lags are defaulted to one time unit. This provides the user with a basic understanding of the system dynamics inherent in the map. Various scenarios are considered and simulated in the map, and are used to brainstorm a preliminary list of problems. After a suitable list is developed, students are asked to list 25 problems, users are asked to state the single root cause of all the problems. This analysis is then summarized in a second written assignment and oral report. Dialectical inquiry then is used to test the strategic analysis and refine the statement of the underlying problem. This then becomes the basis for the third and final stage (see Figure 2).

In the third phase of the study a group list is developed which insures a broad coverage of the emerging technologies. The list is prioritized by the group and 5 hot technologies are agreed to. Each team is then asked to brainstorm how each of the 5 hot technologies could be a creative solution to their company’s strategic problem. This engages the users in a creative thinking process that encourages innovative thinking and problem solving. Each team then selects its top three ideas and develops them as alternative strategic solutions.

Once again the causal map is used to identify how the solution affects the situation. Now the map is revised to illustrate the affect of the solution on the situation. Each solution can have several potential impacts on the situation, it can redesign the map in changing the direction of linkages, it can add or eliminating map nodes, or it can refine the map in changing the intensity of the change coefficient or time lags on the links. This is an aid to understanding the impact of each solution on selected goals and the solutions are then ranked and prioritized according to this analysis.

Two basic approaches to CSM were also explored. The first approach utilized a classic group model-building approach similar to that specified by the systems thinking literature. Causal maps were only developed cooperatively at the group level. This reduces problems of convergence that follow a divergence activity where maps are first developed individually and then merged. This convergence problem is widely known and reported in the GSS literature. One team was specifically asked to first develop individual maps and then make them converge into a single team map (see sample individual map - Figure 3). The initial survey stage utilized a group divergence activity followed by an organizing activity with a final fifteen factors chosen. Fifteen factors is the maximum recommended set of factors for CSM analysis. The most noticeable result from this approach is the difficulty the team had in the divergence phase in arriving at a consensus on the group map. Team members became entrenched in their positions about the fundamental correctness of their view of things requiring a significant amount of inquiry into underlying assumptions and reasoning. This had the effect though of causing the group to come to a refined view of the case as illustrated in the final map (Figure 4).
6. Conclusion

This study has developed both a theoretical foundation and a practical approach to empirical testing of facilitation scripts for engineering dialectical inquiry. This follows a classic usability study as developed for software testing but applies the principles to testing the usability of specific alternatives for facilitation in systems thinking and in particular with a causal mapping approach. Clearly revealed in the study are the following implications:

Dialectical inquiry is an iterative process that assists in revealing and refining underlying assumptions.

Initial conceptual models can be developed and then refined which deepens the analytical thinking and knowledge of group members.

Systems thinking scripts can provide an excellent source of material for developing causal models.

Collaboration engineering precepts provide a structured approach to the overall development of a facilitation model for CSM.

Collaboration engineering provides specific ThinkLets that structure the GSS component for the process of systems thinking.

The basic format of a usability laboratory for the study of CSM facilitation has been established and can provide useful insight into refinement of both the software as a tool and the facilitation scripts.

The next phase of study should move to the integration of a GSS system instead of the current manual process.

The study has now finished an initial exploratory testing. Standard tasks have been developed and basic approaches have been tried and accomplished. This testing has provided guidance on the design of high-level tasks. The comparison testing of several alternate prototypical facilitation scripts have resulted in useful high-level design innovations. The next phase of the study is to do more assessment testing. A deeper-level testing of user tasks that collects quantitative data on specific system functions needs to be developed. Incorporation of standard GSS ThinkLets using Group systems will be developed. This phase will also require a refinement of the software tools as well. Innovation in the facilitation approach also calls for innovation in the software support. This will then allow for verification testing. Prior to final field study use, we will need to verify that the system meets all functional requirements and make minor adjustments to the product before use in corporate settings.

7. References


