

Supporting Collaborative Analysis and Design with Hypertext Functionality

Albert M. Selvin

Network Systems Advanced Technology, Bell Atlantic Corporation,
500 Westchester Ave., White Plains, NY10604, USA

Email: selvin@basit.com

Abstract

Many research efforts and several commercial products have attempted to harness the potential of argumentation-based hypertext tools to enhance the design process. Yet few researchers have reported successful applications of this technology in industry settings. This paper reports on Conversational Modeling, a technique that has been used successfully in a number of collaborative systems analysis, requirements gathering and work process redesign projects in a telecommunications company. The success of the technique appears to be due to the combination of hypertext functionality, a structured modeling framework, and group facilitation strategies. The combination of these elements addresses some of the difficulties reported in the design rationale literature with regard to argumentation-based hypertext tools.

Keywords:

issue-based information systems (IBIS), hypertext functionality (HTF), design rationale, computer-supported collaborative work, business process redesign, argumentation

1 Introduction

As a firm with many internal projects the subject matter and membership of which cross disciplines and departments, Bell Atlantic had a need to provide better ways for distributed teams to organize, analyze and reuse their project information. Several years of experimentation and trial use gave rise to an approach now called Project Compendium, a system that knits together off-the-shelf tools and documents to create a customized environment for collaborative project work. The system provides the capability to convert the elements of various types of documents (email, word processing, spreadsheets, etc.) into nodes and links in a hypertext concept-mapping tool, create associations between those elements while preserving the original set of associations in the source document, and export the associated elements in new documents. In addition, Project Compendium prescribes a set of techniques that can be used within the off-the-shelf tools themselves to add keyword coding and labeling schemas that allow the cross-referencing of ideas and elements from the different sources within the hypertext tool, as well as a set of representational forms that allow groups to do collaborative modeling using hypertext representations. One of the major components of Project Compendium, a modeling technique named Conversational Modeling, has been described in [1, 2, 3].

The system supports a wide range of project activities, including issue-tracking, modeling, planning, analysis, design and other project management tasks. The system also supports facilitated group planning and modeling sessions, conducted either face-to-face or over a network. To use the system, groups define the types or categories of information they are interested in and then create a number of 'templates' embodying these categories that can be used by the various tools involved. Documents loaded into or created within the system are then represented as collections of nodes and links corresponding to the relationships between individual ideas (for example, each paragraph in an email is a separate node, linked to a node representing the subject of the email). Once available to the hypertext database in this manner,

the individual nodes can be linked to any other nodes in the database through associative or transclusive links. For example, individual points in a meeting minutes document can become 'action item' nodes that then reappear in lists of assigned action items to project members, elements of design models and items in a test plan.

The Conversational Modeling approach to collaborative analysis, a key component of Project Compendium, facilitates both formal model-building and unstructured, exploratory, and informal conversation. The approach is unique in the degree to which it integrates both dimensions in a single software environment. Conversational Modeling is concerned with supporting people's ability to construct *meaning* from models. This ability comes from active engagement with both *model-building* activities and with *interpretive* activities -- the consideration of, reflection on, and communication about the models. A key assumption is that useful meaning is not inherent in models, but must be constructed by the model's users -- the team.

This paper presents an argument that many of the issues that have hindered the success of argumentation-based hypertext tools can be addressed by the types of support that Conversational Modeling provides. To do this, we first present the background and set of business problems that gave rise to our early use of argumentation-based hypertext tools. We then cover how our initial problems were mirrored in the research literature, followed by a discussion of potential solutions to those problems. Next, we provide an overview of Conversational Modeling itself. Finally, we focus on how Conversational Modeling's hypertext functionality appears to embody many of the solutions proposed in the literature.

2 Background

Conversational Modeling arose out of our experiences with participatory work systems design projects in a large telecommunications company during 1991-1994. [4, 5, 6] The projects concerned efforts to redesign business processes to achieve better performance, quality and efficiency. Typically, the projects would involve cross-functional teams of five to ten subject matter experts (engineers, technicians, managers, and other personnel), facilitated by one or more members of our staff. During these projects we experimented with a variety of tools and approaches for team modeling and simulation. These experiences revealed a need for a simple and flexible yet robust modeling technique that could be applied to a wide range of project and team situations. Development of Conversational Modeling began in 1993, and the technique has been applied in more than 20 actual work projects in the company since 1994. [7, 8, [added reference](#)]

The participatory work systems design projects attempted to allow cross-functional teams to redesign work processes in an environment characterized by both new and entrenched legacy information systems, multiple stakeholders, compartmentalized 'smokestack' operations, and a fast-changing technology, regulatory and competitive environment. Early experience with participatory design and tools to support such activities, while promising, raised a number of issues. In the course of analysing a problem domain, design teams and their facilitators would generate enormous amounts of information, usually in the form of marked-up easel sheets, containing the results of process mapping, brainstorming, or project management and planning sessions. While many of the individual sessions were insightful and productive, the cumulative effect of the paper-based method was to produce an 'information system' that was unmanageable. Retrieval of desired items or kinds of information became a difficult and time-consuming affair. A strength of the paper-based approach, however, was the ability to shift rapidly between modes of discourse - for example, from structured process mapping to unstructured discussion or brainstorming. Much of the informal, unstructured discussion, however, was not captured, much less made available for later analysis or review.

In addition to the paper-based work of the design team, the early projects employed advanced computerized process modeling and simulation tools. These tools were operated by computer scientists and others working apart from the design team, who relied on the design team for some of their data but built the models in isolation. When the models were presented to the design team for analysis, the

resulting discussion did not get explicitly represented in or linked to the models. The tools employed, while powerful, were so complex that they did not allow for direct or meaningful design team participation apart from acting as providers of data or reviewers of the completed models.

In most cases, neither the paper-based or computer methods recorded the rationale behind design or modeling decisions, so that only the memories of the model-builders or design team members were resources for later recall of the rationale. This proved to be a critical limitation. As the work redesigns were presented to multiple stakeholders at different points of the project lifecycle, project team members were hard-pressed to recall and articulate all the dependencies and rationale for the designs. In addition, as time passed and new members came and old members left the design and facilitating teams, much of the rationale information was essentially lost.

We experimented with a networked issue-based information system (IBIS) hypertext tool which held out the promise of addressing some of the issues above. While promising, our initial experiences revealed that the tool did not provide solutions in and of itself. Early users of the tool produced long, unusable strings of positions and arguments. Maps quickly grew large and unmanageable. Attempting to use the tool to capture the discussion during a meeting often resulted in failure, as inexperienced facilitators attempted to keep up with the rapid conversation and were soon so far behind that they gave up. However, the tool appeared to hold out the potential to help address the 'soft' aspects of team modeling. These aspects include development of group understanding of both models and the problem domain; improving the communicative ability of both the project team as a whole and of the individual members of the team; the storing of rationale for key decisions about model elements; model construction; project management; and other areas of a project. [9, 10]

3 Issues with argumentation-based hypertext tools

Some of the problems that our early users of argumentation-based hypertext tools experienced, such as the rapidity with which users can create confusing and hard-to-navigate information spaces, have been observed in many contexts in the research literature. "One problem with carrying out discussions in the construction space is that it quickly becomes cluttered. It does not take many notes to obscure the design artifact." [11] Conklin [12] identified the 'lost in hyperspace' problem with hypertext tools, and later researchers cite the problem of 'disorientation and cognitive overhead' as hypertext 'disadvantages'. [13] As design information spaces grow large, which happens quickly in intensive process redesign projects, the need for an organizing scheme grows apace. Oinas-Kukkonen cites Yakemovich and Conklin in pointing out that "one of the major drawbacks of many design rationale (DR) methods is the missing macrolevel organization of the hyperspace". [14, 15]

Other researchers have observed that users experience dissonance between the activity of 'doing design' and the need to surface and record design rationale. Buckingham Shum [16] cites "the difficulty of representing useful design rationale while engaging in artifact construction, rapid testing and changing of the [design] artifact, coupled with a reluctance or even inability to interrupt and articulate one's process" results in either incomplete design rationale or incomplete design, as well as some degree of user (designer) frustration. He goes on to invoke Schon's concept of "knowing-in-action" in characterizing skilled design as "spontaneous, skillful execution of the performance" in which designers "are characteristically unable to make [the rationale for their actions] verbally explicit". Conklin and Begeman note that "it is somewhat unnatural to break one's thoughts into discrete units", an attribute of representing complex information domains in the node-and-link parlance of hypertext, "in particular when one doesn't understand the problem well" [17] - as is certainly the case of cross-functional design teams in the early stages of a complex process redesign effort.

Another area of concern is the large degree of time and effort involved to capture and represent design rationale, often involving third parties and considerable expense. Olson *et al.* [18] point out that "trying to capture the design rationales of our meeting discussion takes an enormous amount of coder time off line." Conklin and Yakemovich [19] reported that the IBIS approach seemed to work in actual project settings

only with a scribe taking an enormous amount of time to capture and analyse rationale information. Fischer argues that "the costs of invested effort [using design rationale tools] exceeded the immediate payoff", leading to user resistance and contributing to the "years of [failed] real-world and student projects" using IBIS-based approaches [20]. The time issue alone would prevent design rationale tools from taking root in industry settings characterized by tight deadlines and limited resources.

Many argumentation-based hypertext approaches have gone to some pains to elaborate formal structures for precisely representing design rationale, mandating that users make and record semantic distinctions in the course of design. Much of the literature contains arguments for the relative merits of various rhetorical models, such as QOC [21], QAR [22], PHI [23], DRL [24], variants of IBIS, and others. But, as Buckingham Shum [25] points out, many users have found that, in practice, the sophistication of the distinctions that the structures provide also means that they are too complex or confusing to use, particularly in applied design settings (see also [26]). Our early experience was in accord with this observation; most neophyte facilitators found choosing between the various semantic link types (Expands On, Specializes, Responds To, etc.) and deciding what type of node to create for a particular statement (Issue, Position, Argument) in the course of team meetings to be an insurmountable obstacle. To borrow a concept from human factors engineering, the "affordance" [27] of IBIS and other formal DR structures seems limited outside laboratory settings, despite their formal appropriateness for characterizing design.

Finally, some researchers have commented that it is a leap of some distance to characterize design as a process of arguing between design alternatives. While argumentation does occur, it is but one of many types of communication in the design process. Argumentation and decision-making can be seen as only a subset of design team activities that need support. As Weick and Meader observe when writing on group support systems (GSS), too many GSS focus on decision-making, when sense-making (the process of constructing "moderately consensual definitions that cohere long enough for people to be able to infer some idea of what they have, what they want, why they can't get it, and why it may not be worth getting in the first place") is needed [28]. Certainly, process redesign teams in industry settings usually need to do a great deal of sensemaking work before they can propose and argue over alternatives, and that work needs explicit support that is lacking in many DR approaches.

4 Addressing the difficulties

Researchers in the DR field have identified a number of factors that can address the problems noted in the previous section. A major recommendation is to remove the need to interrupt the flow of design work to articulate design rationale, by integrating argumentation in the process of construction [29]. This can be accomplished by integrating the DR activity into the real work practice of designers. Reeves and Shipman [30] observe that "design rationale should not exist as a separate entity from the design but should be able to exist in the design it is discussing." A commercially available computer-aided software engineering (CASE) tool has incorporated the capability to record hyperlinked IBIS argumentation in the midst of data or process models [31]. Other tools have integrated hypertext argumentation into design applications for computing networks [32], kitchens [33] and chemical plants [34].

Other analysts have pointed to the importance of user buy-in to the use of argumentation-based design rationale tools. [35] Team members must like the approach and be willing to use it. It must provide "rewards" [36] for use in terms of providing appealing and useful functionality. So that "participants become responsible for the creation, maintenance, and continuing relevance" of the design memory, users must be engaged enough with the tool and the process that recording design rationale becomes part of the team's "design culture" [37].

Providing an appropriate language for modeling and design representation can address the problem of overly complex DR schemas. The language must be both accessible to non-specialist users and unproblematically embedded in the software tool. As Stahl writes, "software to support collaborative design should provide a language facility for designers to develop a formal vocabulary for expressing their ideas, for communicating them to future collaborators, and for formally representing them within

computer-executable software. An end-user language is needed that provides an extensible domain vocabulary, is usable by non-programmers, and encourages reuse and modification of expressions." [38]

The software tool in use must provide functionality that allows users to manipulate the design database in a variety of ways without the need for specialists or programming. A principal requirement is the ability to rearrange and combine elements of design models and discussions in many different ways and at many different levels of abstraction, according to the needs of the team at different points in their work. For instance, the Aquanet system provided users with the ability to "see and manipulate a global view of the [issue] network" as well as elements of individual knowledge structures [39]. The system developers observed that "much of the interaction among collaborators takes place through actions on the knowledge structures as well as through meta-comments attached to the knowledge structure." Systems that embed argumentation directly into design applications, such as the network analyser, chemical plant and kitchen design applications mentioned above, provide their users with the ability to act directly on the design materials as well as to record or access rationale information. Some systems provide flexible modes of annotation [40] and a variety of filters that use "different perspectives to filter out the nodes." [41] Others facilitate the interweaving of informal material with IBIS structures [42, 43].

5 Description of Conversational Modeling

Conversational Modeling is the technique that underlies, and gave rise to, the Project Compendium approach. We argue that it embodies many of the solutions proposed above. This section provides a brief overview of the Conversational Modeling technique in order to provide context for that argument. Extensive formal and technical descriptions of Conversational Modeling can be found in [44,45], while reports of user experiences are provided in [46, added reference].

In Conversational Modeling, teams build models by asking and answering questions about aspects of the problem domain and drawing relationships between the answers (as well as asking additional questions). Models consist of the represented questions, answers and relationships. Interpretive activities follow the same schema, and their artifacts are interwoven with the artifacts of model building. The technique aims not so much at building the most complete and accurate model possible, although users have been able to build quite large and comprehensive domain models, but rather at building the best possible understanding and communication both within and around the design team [47].

Conversational Modeling rests on a unique combination of three key elements: customized hypertext software ¹, group facilitation strategies, and a modeling framework ². The software tools include specialized hypertext approaches that enable modeling. The facilitation strategies combine a set of skills that address both task and group maintenance aspects of team modeling efforts [48, 49, 50], as well as providing a key mediation role [51] related to use of the software. The modeling framework creates a structured approach in the development of analysis and design artifacts, and is described below.

5.1 Hypertext functionality

Conversational Modeling prescribes two interrelated components embedded in an issue-based hypertext format. The first of the components is a set of specialized maps constructed according to particular rules. There are three types of specialized maps:

- models, which are representations of aspects, as defined in the modeling framework, of a problem domain;
- catalogs, which collect coded model elements in list form;
- other map types, such as project management overlays and general discussion maps.

The second component consists of techniques for building and linking elements of maps, such as hyperlinking strategies, node coding, templates and labeling conventions. At present, there are no software mechanisms to enforce model-building rules or provide consistency or coherency tests. It is up to

a team's facilitator and to individual team members to ensure that models are kept 'clean' as well as to monitor the overall size and shape of individual maps.

An easy-to-use interface provides a useful set of functions but hides the complexity of the hypertext engine from the user. In brief, the features of most importance here are:

- The ability to create hyperlinks by copying a node from a map or list and pasting it into another (there is, in fact, no 'copy' made, but rather the same node is now present in more than one 'view'). Any change to any appearance of a node is immediately reflected in all other appearances of that node.
- A Containing Views feature that allows the display of a scrolling list showing all the views (maps or lists) that a particular node is 'contained' in. The scrolling list is also a set of hyperlinks; users can click on any of the views listed and the system will take them to the node of interest in that view.
- Each node contains a Contents Window that allows users to enter additional information about the node, including keywords that can be used in searches of the database. Contents windows are made visible by double-clicking on the node.
- A search engine that can display nodes containing certain keywords in scrolling lists, that can then be sorted according to various criteria. The lists are themselves sets of hyperlinks, as users can employ the Containing Views feature to locate the nodes of interest in the various contexts (maps or lists) those nodes appear in.

5.2 Modeling format

The modeling technique makes use of IBIS conventions [52]. The activity of model-building, and the models themselves, follows a question-and-answer format. The questions are drawn from defined templates for each model type, while the expected answers conform to the rules established for that model type. Figure 1 shows the general form of model elements in this technique. The questions and templates were derived from structured modeling approaches and from experimentation and discussion with groups engaged in modeling. For example, in one project which used Conversational Modeling, questions about objects and their attributes were inspired by formal object-oriented analysis [53, 54] while questions about organizations and roles were originally based on elements of the Common KADS Organization Model [55, 56]. Questions about problems and opportunities were generated by the facilitators, in response to issues raised by the participating groups, and by members of the groups themselves.

Figure 1

Figure 1. General form of models in the Conversational Modeling approach

It should be noted that the predefined templates are *suggestions* for which combinations of questions and attributes 'belong' to model elements for the various model types. How closely a particular team must hew to the templates is a function of the time available, goals, preferences and skills of the participants. Adding ideas, attributes, or questions that don't 'fit' with a predefined template but which are important to the team during a session is part of the value of the approach.

5.3 Types of models

There are eight types of models in Conversational Modeling, corresponding to those in the World Modeling framework (Figure 2). [57] Five of these types have been used most extensively in project work to date, and are described here. The models are grouped into four quadrants representing different views of the problem domain: the *current implementation* quadrant, which models how the work system is currently implemented in the organization; the *current essential* quadrant, which presents an abstracted view of the work system; the *future essential* quadrant, which models the future (designed) state of the work system in the abstract; and the *future implementation* quadrant, which provides a representation of the concrete future state of the work system.

Figure 2

Figure 2. World Modeling framework ([full-size figure](#))

Each quadrant contains a number of individual model types, along with several model types common to all the quadrants. Several of these are described here (other model types include knowledge, timing, object, process, geography and distribution):

- **Organization models** describe the social structure at work in a problem domain. They are intended to allow participants to gain understanding about the roles that different groups and individuals play in the problem domain. Both formal and informal organizational roles and relationships can be represented as Figure 3 illustrates for a particular example system.
- **Communication models** show the formal and informal communication between roles. They may also be used to represent the communication between human users and computer systems, or between systems and other systems.
- **Task/activity models** represent the sequences and attributes of tasks and activities in the problem domain.
- **Resource models** represent the attributes of people (as individuals, not as organizational roles) or physical resources involved in the problem domain. There are many kinds of resources, including systems, tools, paper forms, electronic parts and vehicles.
- **Problem & opportunity models** (Figure 4) are places where the issues that have come up in the course of modeling other aspects of the problem domain are discussed and analysed by the project team. These models most closely resemble traditional IBISs, since the act of identifying and representing problems and opportunities in the problem domain often leads to proposals for alternative solutions and attendant argumentation. Issues that arise in the course of reviewing and discussing Problem & Opportunity Models may also serve to inform subsequent model-building activities.

There are a wide variety of other specialized map types that support Conversational Modeling. The flexibility of the technique allows for new types of useful views to be defined on the fly, with full inter-linking capabilities to existing maps and lists. One type of specialized map that has proven useful is project management overlays. As with modeling activities, project management represents an area of project work where free-form, exploratory discussions alternate with and inform structured activities (and vice-versa). These activities include defining schedules, deliverables, project tracking and status reporting. Linking exploratory discussions about project management issues with the results of those discussions (for example, linking discussions about the topic "What should our deliverables be?" with maps showing the agreed deliverables), as well as with artifacts of actual project work (e.g. models), is helpful in integrating the various strands of a project. Other map types include issue-tracking views, requirements management maps, deliverables (maps that were presented to project clients, usually after being converted to text and incorporated in a document), and discussions (free-form IBIS conversations).

Figure 3

Figure 3. An organizational model ([full-size figure](#))

Figure 4

Figure 4. Task/activity model ([full-size figure](#))

5.4 Hyperlinking strategies

Effective hyperlinking allows the creation of cross-references, multiple views, and a rich and robust model set. The more it is useful to understand an aspect of a problem domain in multiple dimensions, the more it is useful to create hyperlinks. A node that has a certain meaning in a specialized map or list can play different roles in other maps or lists. Hyperlinking shows, reinforces and manages this multidimensionality of meanings. The other techniques discussed in this document are in support of, and achieve their true leverage through, these hyperlinking strategies.

An example: in modeling a role named Technician, the team has identified Service Order, one of the information objects that the role comes into contact with, as one of Technician's attributes. Service Order is itself modeled on an Object Model map, with its own attributes, including a Problem/Opportunity attribute titled 'Not implemented correctly'. Service Order also appears as an object attribute on a Task Model map for a task named 'Installation'. The object also appears as an entry in a Catalog of Objects. Finally, the team has identified Service Order as one of several objects for which there are 'Implementation problems (in a Problem/Opportunity model map).

The node named Service Order appears in all four maps as the result of copy-paste actions (i.e. the team has copied the node and pasted it into each map). When a user selects the Containing Views option while pointing to one of the appearances of the Service Order node, the software will tell them all the other places that Service Order appears. This aids in identifying all the contexts in which Service Order is relevant, which helps the project team understand the implications and dependencies of design changes.

5.5 Hyperlinking in practice

The general procedure for hyperlinking is to create a link (i.e. paste an existing node into one or more additional maps) whenever there is a relationship that the team believes is or may become significant to understanding the problem domain, managing the project, or communicating effectively.

This kind of hyperlinking can, in itself, produce new understanding. By continually asking the question "where (else) should we link this element?" the team, in effect, is asking "where is this piece of information significant?" Thus, the team is continually interpreting the significance of the following to the goals of the project:

- individual model elements
- relationships between elements
- relationships between model

The extent and volume of hyperlinking that a team undertakes should be driven by the project's goals and the time available for modeling, along with other factors. There are always trade-offs between model extent and model usability.

5.6 Node coding

Node coding involves placing codes in the Detail field of the Contents window of particular nodes. The codes enable searches, cross-references and on-the-fly creation of specialized maps and lists without the manual work of assembling them node-by-node. Since subsequent searches will find coded nodes wherever they may lie in the database, teams effectively 'seed' other model types and catalogs by entering a code whenever they create a node of one of the specialized types.

5.7 Labeling conventions

To ensure that entries on maps, lists, catalogs and hyperlinks support effective browsing, searching and perusal, a number of labeling conventions should be adhered to. These conventions apply to some node and map labels and descriptions. For example, in maps containing models of individual roles, it is recommended that the roles be labeled with the name of the role followed by the word "(Role)" in parentheses. Thus subsequent Containing Views selections and Catalogs will be easier to read and understand.

5.8 Templates

Templates are a way to facilitate the creation of model maps or other specialized maps that follow a structured format. The formats consist of a set of questions that are asked of model elements. These

questions can be saved in templates and brought in to new model maps. For instance, the process of modeling a role can be jump-started by importing the Role Model template.

6 Addressing argumentation-based hypertext issues

The next two sections present the argument that Conversational Modeling's combination of techniques addresses many of the issues raised in sections 3 and 4.

Conversational Modeling integrates DR into the work practice of design teams by incorporating it into facilitated domain analysis sessions. By employing a model-based approach to design, the act of building models in the various quadrants of the modeling framework continually places teams in the position of asking and answering a variety of questions about the problem domain. These questions, which correspond to the evolving model templates, are designed to elicit and integrate domain understanding, design elements and design rationale without forcing consideration of these areas into separate activities or tools. Facilitators play a key role by helping teams understand and work with the tool and modeling framework, as well as by paying attention to "individual personalities, emerging group norms, and political realities" [58] and ensuring that conditions are suitable for continuing development of shared understanding among the team.

The design of Conversational Modeling was heavily influenced from its inception by principles from the Participatory Design literature. [59] Early work on the technique was motivated, in part, by a desire to allow telephone company employees to participate as deeply as possible in the knowledge engineering of a planned capital budgeting expert system. For a technique to be embraced by its intended users, those users would need to see the value of the technique and enjoy using it.

To evaluate the effectiveness of Conversational Modeling for its users, an informal study was conducted, consisting of interviews with approximately 25% of the participants in the four projects using the technique in 1994 and 1995. [60] Each interview lasted 30-60 minutes. Participants were asked to describe their experience with Conversational Modeling; the effect, if any, on the ability of the people involved to learn and communicate about the subject matter; and the technique itself. Participants ascribed a high degree of value to the technique, reporting that it helped their teams to gain a higher degree of shared understanding and insight than other techniques they had used in their current or previous projects. Many reported that Conversational Modeling allowed them to gain both deep and broad understanding of their problem domain:

At first, to be honest, I found it frustrating because it would be asking all these questions. You thought you were through, out would come another question. Later, as things tied in, it would seem like a seed of a big plant. You took from beginning to end, got the big picture, then you realized the reasoning behind the technique: to gather as much information about an organization so you had the whole picture so one thing feeds another.

Many participants observed that the technique played a positive role in the formation, communication and group process aspects of the teams using it, especially in the early days of the team's existence. In particular, the technique appeared to enhance mutual respect, attention and learning. Respondents reported that the technique helped enable their teams to build and understand complex models more quickly and with greater comprehension than other techniques. Such comprehension may result in better, more integrated designs and more effective implementations. For some of the projects studied, Conversational Modeling enabled teams to produce deliverables by combining elements of model-bases they had built without needing to use other software tools or perform new analysis, thus saving time and maintaining potentially useful direct links between discussions, models and deliverables. Finally, participants reported that Conversational Modeling helped their teams achieve both clearer models and a shared sense of purpose and direction:

I miss it very much. It was like a guide, it led you and you followed. It beckoned you and you followed. It prompted you. Whereas with a lot of other things we've used we

spent a lot of time trying to understand what we're referring to. They're not as clear-cut, not as definitive.

While Conversational Modeling recognizes and supports the need for formally represented argumentation for design alternatives, such argumentation is neither the center nor the essence of the technique. To provide maximum flexibility for domain modeling purposes and to provide maximum accessibility and ease of use for design team members, Conversational Modeling moves the focus from argumentation to framing, re-framing and organizing elements of the problem domain. In a sense the technique embeds what might be called the *inherent argumentation* of domain modeling in the activity of using the technique: every time a design team makes associative or transclusive links between model elements, they are inherently making an 'argument' about what relations exist in the domain and how they are important. The models thus become a forum, repository and visual embodiment of the team's collective determination of what is significant about the domain for their purposes.

Conversational Modeling avoids the complexity of many DR frameworks by removing much of their semantic intricacies, such as varied link types (although users can choose to employ the types if they desire). Conversational Modeling uses simple schemas (such as the templates) requiring as few rhetorical abstractions as possible. The team's abstractions are in the domain modeling framework, not the representational schemas. The technique trades refinements of meaning that can be useful for later DR purposes for an emphasis on developing a critical mass of team-produced artifacts (such as models) and the ability to generate associations rapidly. We believe that having to choose link types or create explicit argumentation (pros and cons) slows down the modeling process and adds cognitive overhead for users.

7 Addressing the issues - hypertext functionality

In combination with the facilitation strategies and the modeling framework, hypertext functionality (HTF) [61] provides sufficient affordance for the users of the Conversational Modeling approach. Details of the uses and benefits of the team's HTF are covered more comprehensively in [62, 63]; this section discusses a few examples of how the HTF addresses the difficulties and recommendations presented in the first half of this paper.

Effective "macrolevel organization of the hyperspace" [64] is accomplished by integrating the modeling framework into the hypertext database. The simple, coherent and repetitive structure of the modeling framework, as implemented in templates, node coding, node labeling, and the other hypertext strategies employed, helps to generate exploratory thinking while ensuring that both model elements and related discussion are always contextually grounded and locatable in the database. Organization of the database according to the modeling framework addresses the 'lost in hyperspace' phenomenon by giving users an abstract, domain-independent (and team culture-independent) framework that is consistent both within individual maps and between all maps in a database. HTF aids the relational and multidimensional aspects of the locational framework by allowing teams to integrate models with each other, but avoids 'spaghetti' hypertext by mandating that all links are created in conformance with the rules prescribed by the modeling framework.

The team's HTF permits and encourages what Buckingham Shum [65] calls "opportunistic leaps" in modeling by making it easy to create and follow associative and transclusive links between model elements. Teams can make connections and associations and represent them permanently and visually in the database through such linking activities. For example, the simple act of adding a Role code to a node labeled Switch Engineer indicates that the team has made several layers of associations: first, that the concept of Switch Engineer is a role within the organization that the team is concerned with; second, that this role is significant enough within the problem domain to be both modeled on the present map and made available through searches and other means for subsequent linking and modeling.

Hypertext makes possible the rapid modification and extension of the team's modeling framework, principally through modification of the modeling templates and extensions to a project's coding scheme. As a team becomes more immersed in and familiar with both their

problem domain and the activities associated with Conversational Modeling, they reflect on the composition of the templates and make changes reflecting their concerns. This activity can play an important role in deepening team engagement with their analysis of the domain. As the team's understanding of the task changes...the evolution of structuring schemes is an important side-effect of using them." [66] Such changes will be manifested in subsequent importation of the template into models, thus guiding the questions and answers asked in relation to a model element. In fact, IBIS-based discussions can be held about the composition and use of the templates and node coding scheme, and the maps recording such discussions can be valuable sources of project management rationale.

Conversational Modeling's HTF plays a key role in providing rewards to its users. A key feature in this respect is the ability to 'seed' models and catalogs on the fly. For example, as users build a model of the Switch Engineer role, they populate the template with attributes of that role in the form of coded nodes, such as objects like Capacity Utilization Report, Switch Type and Remote Office. By virtue of placing the 'Object' code in the contents of these nodes, subsequent Catalogs of Objects will include the Switch Engineer's objects without any extra manual linking work. The team can then begin to build their object model using the elements of the Catalog of Objects, copying nodes to populate the attributes of each object from other, similarly created Catalogs (such as Catalogs of Tasks, Roles, or Resources). Similarly, by placing an 'open issue' code in the contents of a node, an up-to-date Catalog of a team's open issues can be produced at any moment. The node labeling scheme provides a number of similar rewards, such as facilitating rapid queries. For example, a team can right-click on the Switch Engineer node to display a list of views that the node is included in. This allows the team to see that the Switch Engineer is present in the communication models of three other roles, indicating that communication between the Switch Engineer and other roles is a potentially significant element for subsequent analysis and modeling.

8 Conclusion

Teams charged with designing effective business process or technology interventions need access to all the rich detail about their problem domain without being overwhelmed by its complexity. They gain their understanding of the 'big picture' through understanding the meaning of the elements of the domain, at many levels of granularity. Such meaning is not inherent in the elements themselves; it must be produced through collaborative analysis and conversation. Ideally, tools could support each mode (modeling and conversation) with full hyperlinking capabilities as well as a rich set of representational possibilities. Based on off-the-shelf hypertext technology, the technique described in this paper, of necessity, relied on a limited representational palette and other restrictions in representational and hyperlinking capabilities. Future efforts could expand on either side of the equation to provide more powerful tool sets for team analysis.

For all their potential, most argumentation-based hypertext approaches have not proven as useful as hoped in industry settings. Through its combination of three key elements—a well-integrated modeling framework, the use of skilled facilitators and clear facilitation strategies, and a flexible, multi-dimensional hypertext tool and set of prescribed linking, coding, and labeling techniques—Conversational Modeling has both gained acceptance by users and been used effectively in projects. Each of the three key elements appears to be critical to the success of the technique, especially in combination, as each element buttresses the strengths and limitations of the others. When combined with an IBIS-based DR approach and applied with care, Conversational Modeling holds out the potential for issue-based hypertext to realize the value foreseen by researchers.

Acknowledgements

The author thanks Bell Atlantic Network Systems Advanced Technology, in particular Jim Euchner, Howard Citron and Neil Friedman, for supporting the Project Compendium and Conversational Modeling work. In addition the author thanks Maarten Sierhuis (NASA Ames Research Center), Jeff Conklin (GDSS Inc.) and Simon Buckingham Shum (Open University, UK) for the many conversations and

collaborative efforts that gave rise to the ideas in this paper.

Notes

¹ For the work described in this paper, the software tool employed was QuestMap, developed by Corporate Memory Systems, Inc. and now marketed by GDSS, Inc.

² Until 1997, we used the term \0xFFFDConversational Modeling\0xFFFD to describe this combination of elements. In 1997, we began to develop our own software tool set and provided training in the overall approach, which by then included explicit support for activities such as project management and distributed collaboration. Since our scope had grown beyond 'modeling' *per se*, we changed the name to Project Compendium. Since earlier published papers refer to Conversational Modeling, we use that term for the present discussion.

References

- 1 **Selvin, A.** (1997) "An Issue-Based Hypertext Approach to Collaborative Process Redesign". Nynex Science & Technology Technical Memorandum TM 97-0007, White Plains, NY
- 2 **Selvin, A. and Sierhuis, M.** (1996) "Experiences with Conversational Modeling: Building Models Through Collaboration". Nynex Science & Technology Technical Memorandum TM 96-0045, White Plains, NY
- 3 **Selvin, A.** (1996) "Leveraging Existing Hypertext Functionality to Create a Customized Environment for Team Analysis". *Proceedings of the Second International Workshop on Incorporating Hypertext Functionality Into Software Systems*, Bethesda, MD, March <http://www.cs.nott.ac.uk/~hla/HTF/HTFII/Selvin.html>
- 4 **Euchner, J., Sachs, P. et al.** (1993) "The Benefits of Intentional Tension". *Communications of the ACM*, June
- 5 **Sachs, P.** (1995) "Transforming Work: Collaboration, Learning, and Design". *Communications of the ACM*, September
- 6 **Kleiner, A.** (1995) "The Battle for the Soul of Corporate America". *Wired*, 3.08, August <http://www.wired.com/wired/archive/3.08/reengineering.html>
- 7 **Selvin, A.** (1994) "Human Interaction Issues with Group Support Technology in a Software R&D Organization". Nynex Science & Technology Technical Memorandum TM 94-0059
- 8 **Selvin, A.** (1994) "Experience with CM/1: Position Paper for 'Solving Groupware Problems' ". Panel, in *Proceedings of Groupware \0xFFFD94 conference*, Boston
- 9 **Checkland, P. and Scholes, J.** (1990) *Soft Systems Methodology in Action* (Chichester: Wiley)
- 10 **Dennis, A., Hayes, G. and Daniels, R.** (1994) "Reengineering Business Process Modeling". In *Proceedings of the 27th Annual Hawaii International Conference on System Sciences*
- 11 **Reeves, B. and Shipman, F.** (1992) "Supporting Communication Between Designers with Artifact-Centered Evolving Information Spaces". In *CSCW'92: Computer-Supported Cooperative Work* (ACM Press: New York)
- 12 **Conklin, J.** (1987) "Hypertext: An Introduction and Survey". *IEEE Computer*, Vol. 20, No. 9
- 13 **Oinas-Kukkonen, H.** (1996) "Hypermedia functionality in modelling tools". In *Proceedings of the Second International Workshop on Incorporating Hypertext Functionality Into Software Systems*, March

14 Oinas-Kukkonen, H., [ibid](#)

15 Yakemovich, K. and Conklin, J. (1990) "Report on a Development Project Use of an Issue-Based Information System". In *CSCW \0xFFD90 Proceedings*

16 Buckingham Shum, S. (1996) "Analyzing the Usability of a Design Rationale Notation". In *Design Rationale: Concepts, Techniques, and Use*, edited by T. Moran and J. Carroll (Mahwah: Lawrence Erlbaum)

17 Conklin, J. and Begeman, M. (1988) "gIBIS: A Tool for Exploratory Policy Discussion". In *Proceedings of CSCW \0xFFD88*

18 Olson, G., Olson, J., Storosten, M., Carter, M., Herbsleb, J. and Rueter, H. (1996) "The Structure of Activity During Meetings". In *Design Rationale: Concepts, Techniques, and Use*, edited by T. Moran and J. Carroll (Mahwah: Lawrence Erlbaum)

19 Conklin, J. and Yakemovich, K. C. (1996) "A Process-Oriented Approach to Design Rationale". In *Design Rationale: Concepts, Techniques, and Use*, edited by T. Moran and J. Carroll (Mahwah: Lawrence Erlbaum)

20 Fischer, G., Lemke, A., McCall, R. and Morch, A. (1996) "Making Argumentation Serve Design". In *Design Rationale: Concepts, Techniques, and Use*, edited by T. Moran and J. Carroll (Mahwah: Lawrence Erlbaum)

21 Buckingham Shum, S., MacLean, A., Bellotti, V. and Hammond, N. (1997) "Graphical Argumentation & Design Cognition". *Human-Computer Interaction*, Vol. 12, No. 3, 267-300

22 Oinas-Kukkonen,H., [ibid](#)

23 Fischer, G. *et al.*, [ibid](#)

24 Lee, J. and Lai, K. (1996) "What\0xFFDs In Design Rationale". In *Design Rationale: Concepts, Techniques, and Use*, edited by T. Moran and J. Carroll (Mahwah: Lawrence Erlbaum)

25 Buckingham Shum, S. "Analyzing the Usability of a Design Rationale Notation". [ibid](#)

26 Halasz, F. (1988) "Reflections on Notecards: Seven Issues for the Next Generation of Hypertext Systems". *Communications of the ACM*, 31(7), July, 836-852

27 Norman, D. (1988) *The Design of Everyday Things* (New York: Basic Books)

28 Weick, K. (1993) "Sensemaking and Group Support Systems". In *Group Support Systems: New Perspectives*, edited by L. Jessup and J. Valacich (New York: Macmillan)

29 Fischer, G. *et al.*, [ibid](#)

30 Reeves, B. and Shipman,F., [ibid](#)

31 Oinas-Kukkonen,H., [ibid](#)

32 Reeves, B. and Shipman,F., [ibid](#)

33 Fischer, G. *et al.*, [ibid](#)

34 Ballinger, G., Banares-Alcantara, R. and King, J. (1993) "Using an IBIS to Record Design Rationale". Technical Report 1993-17, University of Edinburgh Department of Chemical Engineering

35 Reeves, B. and Shipman,F., [ibid](#)

- 36 **Lee, J.** (1990) "SIBYL: A Tool for Managing Group Design Rationale". In *Computer Supported Cooperative Work* (New York: ACM Press)
- 37 **Moran, T. and Carroll, J.** (1996) "Overview of Design Rationale". In *Design Rationale: Concepts, Techniques, and Use*, edited by T. Moran and J. Carroll (Mahwah: Lawrence Erlbaum)
- 38 **Stahl, G.** (1993) "Supporting Situated Interpretation". In *Proceedings of the 15th Annual Conference of Cognitive Science Society* (Mahwah: Lawrence Erlbaum Associates)
- 39 **Marshall, C., Halasz, F., Rogers, R. and Janssen, W.** (1991) "Aquanet: A Hypertext Tool to Hold Your Knowledge in Place". In *Proceedings of Hypertext'91* (New York: ACM Press)
- 40 **Oinas-Kukkonen, H.**, [ibid](#)
- 41 **Reeves, B. and Shipman, F.**, [ibid](#)
- 42 **Haake, J. and Wilson, B.** (1992) "Supporting Collaborative Writing of Hyperdocuments in SEPIA". In *CSCW'92: Computer-Supported Cooperative Work* (ACM Press: New York)
- 43 **Streitz, N., Gebiler, J., Haake, J. and Hol, J.** (1994) "Dolphin: Integrated Meeting Support across LiveBoards, Local and Remote Desktop Environments". In *CSCW'94: Computer-Supported Cooperative Work* (ACM Press: New York)
- 44 **Selvin, A.** (1996) "Conversational Modeling: A Software-Supported Technique for Analysis by Teams". Nynex Science & Technology Technical Memorandum TM96-0013
- 45 **Selvin, A.** (1996) "Leveraging Existing Hypertext Functionality to Create a Customized Environment for Team Analysis". [ibid](#)
- 46 **Selvin, A. and Sierhuis, M.** (1996) "Experiences with Conversational Modeling: Building Models Through Collaboration". Nynex Science & Technology Technical Memorandum TM96-0045
- 47 **Richmond, B.** (1994) "An Introduction to Systems Thinking with IThink". High Performance Systems
- 48 **Clawson, V., Bostrom, R. and Anson, R.** (1993) "The Role of the Facilitator In Computer-Supported Meetings". *Small Group Research*, Vol. 234, No. 4
- 49 **Bostrom, R., Anson, R. and Clawson, V.** (1993) "Group Facilitation and Group Support Systems". In *Group Support Systems: New Perspectives*, edited by L. Jessup and J. Valacich (New York: Macmillan)
- 50 **Freeman, P.** (1994) "The role of Manager as Facilitator: Southern California Edison Experience Using the IBIS Method". *The Facilitator*, issue 5, September
- 51 **Okamura, K. et al.** (1994) "Helping CSCW Applications Succeed: The Role of Mediators in the Context of Use". In *Proceedings of CSCW '94* (New York: ACM Press)
- 52 **Kunz, W. and Rittel, H.** (1970) "Issues as Elements of Information Systems". Working Paper 131, Institute of Urban and Regional Development, University of California at Berkeley
- 53 **Coad, P. and Yourdon, E.** (1991) *Object-Oriented Analysis* (Englewood Cliffs: Prentice-Hall)
- 54 **Jacobson, I. et al.** (1992) *Object-Oriented Software Engineering: A Use Case Driven Approach* (Wokingham: ACM Press)
- 55 **de Hoog, R. et al.** (1993) "Applying the Common KADS Organization Model". KADS-II Consortium (ESPRIT Project P5248)

- 56 **Kruizinga, E. and van der Spek, R.** (1993) Model-Based Development of Knowledge-Intensive Systems Workshop, Centrum voor Kennistechnologie
- 57 **Sierhuis, M., van der Spek, R. and Selvin, A.** (1998) "World Modeling: A Framework for Modeling Complex Systems". Unpublished working paper
- 58 **Niederman, F., Beise, C. and Beranek, P.** (1996) "Issues and Concerns About Computer-Supported Meetings: The Facilitator's Perspective". *MIS Quarterly*, March
- 59 **Greenbaum, J. and Kyng, M.** (1991) *Design at Work: Cooperative Design of Computer Systems* (New York: Erlbaum)
- 60 **Selvin, A. and Sierhuis, M.** (1996) "Experiences with Conversational Modeling: Building Models Through Collaboration". [ibid](#)
- 61 **Oinas-Kukkonen, H. and Bieber, M.** (1994) "Incorporating Hypertext Functionality into Software Systems". *ECHT'94 Workshop* <http://info.acm.org/siglink/ECHT94TR/Functionality.html>
- 62 **Selvin, A.**, "Conversational Modeling: A Software-Supported Technique for Analysis by Teams". [ibid](#)
- 63 **Selvin, A.**, "Leveraging Existing Hypertext Functionality to Create a Customized Environment for Team Analysis". [ibid](#)
- 64 **Oinas-Kukkonen, H.**, [ibid](#)
- 65 **Buckingham Shum, S.**, "Analyzing the Usability of a Design Rationale Notation". [ibid](#)
- 66 **Marshall, C., et al.**, [ibid](#)

Added reference

This reference added on 13/1/99 after the paper had been edited

Selvin, A. (1998) "Supporting Granular Reuse of Knowledge Elements in an Organizational Memory System." In *Proceedings of the Seventh International Workshop on Hypertext Functionality: Organizational Memory Systems & HTF*, Helsinki, Finland