Discourse-Based Techniques for the Analysis of Online Collaboration

Alexander Feinman and Richard Alterman
Brandeis University

August 13, 2004

Alexander Feinman is a cognitive scientist with interests in collaboration and groupware analysis; he is a Ph.D. candidate at Brandeis University. **Richard Alterman** is a cognitive scientist with interests in cognitive modeling, planning and activity, and text and discourse; he is a Professor at Brandeis University.

Corresponding author:
Alexander Feinman
Computer Science Department
Brandeis University MS018
415 South St., Waltham MA 02454 USA
afeinman@cs.brandeis.edu

Abstract

Online collaboration has become very common. Remotely-taught educational courses, collaboration with work divisions in remote locations, and coordinating military personnel distributed across an information-driven battlefield all require design work to construct computer-mediated activities that enable participants to effectively coordinate remotely. Understanding the complex interaction of online participants in a joint activity is crucial to designing effective software tools to support their task. By examining the patterns of interaction participants create we can reveal the effort they expend both to perform their task and to maintain coordination with each other.

We created two methods for systematically analyzing online interaction. One examines coordination issues at a social level by isolating and examining recurring problems in coordination, and by investigating the secondary structure participants create in the discourse to handle these complexities. The other method examines the interaction at an individual cognitive level, using the references participants make in the discourse as a way to infer the cognitive load that various representations of information incur in the participants. These two methods have been applied to examine data from VesselWorld, a groupware test bed, and data collected with a variety of other groupware systems. Using these methods we were able to explain the successes and shortcomings of the introduction of a new representation system. We also collected data from a semester-long experiment in which a class of students ran experiments, collected usage data, and applied the methods to successfully justify redesigns of the representation system within a groupware system to support and simplify coordination.

Contents

List of Figures

1 Introduction

Online collaboration has become very common. Remotely-taught educational courses, collaboration with work divisions in remote locations, and coordinating military personnel distributed across an information-driven battlefield all require design work to construct computer-mediated activities that enable participants to effectively coordinate remotely. However, staying coordinated in these different-place interactions can be difficult. Because they are in different locations and have access to different physical environments, the methods participants use for pointing at, modifying, and reviewing objects, as well as gauging the focus, intent, and emotional state of other participants, are different from those they would use in a face-to-face interaction. Thus, the effective procedures for the maintenance of common ground are significantly altered, even when high-fidelity technology such as video conferencing is used. This means that careful design is required to support online interaction properly.

The challenge of designing easy-to-use systems has been called a "wicked problem" (Rittel & Webber 1984; Fitzpatrick 2003), in that it necessarily involves re-examining the set of potential solution methods for each particular situation. What is needed, therefore, is not just a generalized set of rules about design, but also a process that allows a designer to properly investigate the problem at hand and establish a customized solution path. Frameworks such as iterative design (Gould & Lewis 1985) provide a way to talk about the process of creating systems. Figure 1 shows an expanded view of iterative design. The process involves a cycle of design, deployment, and evaluation that repeats until the design is satisfactory, at which point it is deployed to users. Satisfactory results, however, may require a large number of cycles, a process that can be very expensive or otherwise impractical due to time constraints or availability of test users. In business, long design cycles can lead to premature failure of products (Norman 1998). Hence, it is imperative to reduce the number of revisions necessary in constructing a successful product. The methods presented in this paper focus specifically on improving the evaluation and redesign portions of this cycle.

Creating groupware software to support an ongoing, online, same-time/different-place (Ellis, Gibbs, & Rein 1991) collaboration is a very challenging design problem. Despite the best efforts of designers, groupware applications often end up interfering with the very work they are designed to support (e.g., Foster & Stefik 1986). Ideally, a software system would be built to match the emergent practice of a community of users (Schmidt &

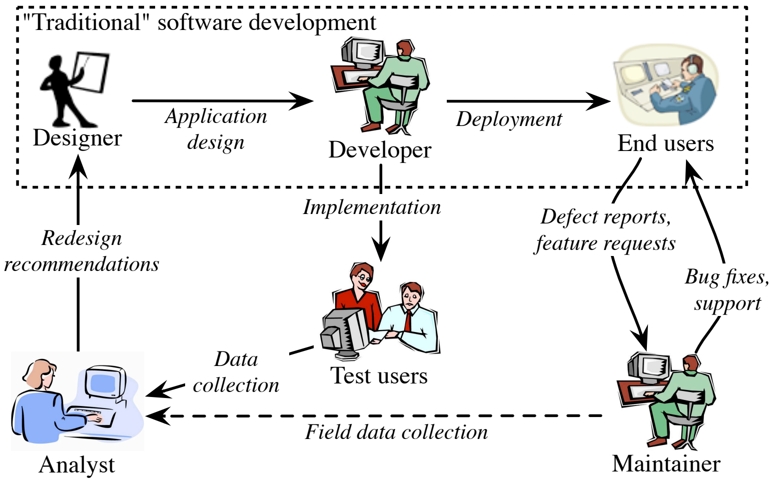


Figure 1: Expanded iterative-design cycle (after Landsman & Alterman 2003)

Bannon 1992). However, creating such a system requires a thorough understanding of that emergent practice.

The introduction of a groupware system into an existing collaboration changes the nature of the interaction (Hutchins 1995b). At Brandeis, theGROUP lab has been developing methods for constructing and analyzing computer-mediated collaborations. One part of the project includes a toolkit for building groupware systems that automatically produce complete transcripts of the interaction and can replay these transcripts. Another part is primarily concerned with analyzing participant interaction in these activities. To better understand the impact of introducing new representations into a collaboration, we have have devised a pair of methods for analyzing interaction data. The first method looks for recurrent patterns of interaction and takes note of the secondary structure that participants create to organize their behavior. The second method looks at the referential structure created by participants as they refer to common objects. Together these methods provide a way to understand the specific ways that introducing new ways for participants to share information can have on an interaction. This paper will explain and examine these methods.

1.1 Representation Systems and Coordinating Representations

In an online, ongoing cooperative activity, sharing context is crucial. Due to the reduced scope of interaction when compared with face-to-face interaction, successful online interaction requires groupware systems that facilitate realignment of common ground. Previous work (e.g., Hutchins 1995b; Norman 1991) has explored a view of the common ground shared by participants of a cooperative activity in terms of the *representation system* available to them. A representation system is made up of three parts:

1. A set of *representational media* available to the participants.
2. A set of *specific representations* available to the participants (internal or external, private or shared, implemented ahead of time or created during the interaction).
3. A set of *procedures* for recording, reviewing, modifying, transcribing, and aligning information between multiple, partial representations of the shared context.

In non-face-to-face interactions, structures that simplify the coordination of a conventional behavior can be codified into artifacts. Past work has examined the role of external artifacts in coordinating interaction (Schmidt & Wagner 2002). *Coordinating representations* (CRs: Suchman & Trigg 1991; Alterman et al. 2001) are ubiquitous coordinative artifacts that present a way for participants to organize their behavior in a joint activity by creating shared expectations of roles and actions and by partially structuring actions. For example, a stop sign creates expectations in the participants of a joint traffic activity but does not determine activity completely. An agenda for a meeting serves both to organize activity by partially ordering topics for discussion and by creating expectations about the structure of the meeting. This mediation can fundamentally change the interaction. Many CRs (such as a to-do list) can be modified as the activity progresses, allowing them to serve as external repositories of information. Others, like the stop sign, are immutable but nevertheless serve to modify the internal representations a participant has for the interaction. In general, CRs serve to simplify a task both by offloading some of the cognitive load of the task (such as the way a notebook serves to ease the burden of remembering information; Perkins 1993) and by altering the task to make problem-solving easier (e.g., aligning multiple sources of information as in "complex sheets" for airport luggage; Suchman & Trigg 1993).

However, providing additional representations to participants in a joint activity can cause problems. Because the view of a joint activity necessarily differs from person to person, participants must align their private representations during the activity to the extent necessary for the activity to be successful. A smooth flow of collaborative activity provides evidence that the participants have similar conceptions of the activity. When participants find that their private representations have become dissimilar to the point where further work becomes difficult, they will employ alignment procedures to restore common context.

A representation system gives participants a choice regarding how to distribute and manage information. Different representations present different costs for recording, reviewing, modifying, transcribing, and aligning different sorts of information. For example, participants at a meeting can decide to communicate ideas verbally or to express them with the help of a whiteboard or other such device. In such a case, the whiteboard may be favored for ideas that are complex, benefit from visual display, or need to be available for discussion at a later time. In these cases, the cost to transcribe the information to the whiteboard is outweighed by the potential benefits. The choices that participants make about how to record, transcribe, etc. information are at least partly influenced by the principle of Least Collaborative Effort (Clark & Brennan 1991); in a long-term, cooperative interaction, participants tend toward matching up information with representations that store it in a fashion that requires the least overall effort.

One reason a groupware system can prove ineffective is that the representation system it offers does not complement the characteristics of the task. That is, the access to information that the coordinative artifacts provide do not match the access characteristics of the task information, and the available methods of coordination do not match the necessary forms of interaction required to successfully perform the task. This mismatch leads to increased work and increased potential for error on the part of the participants as they try to fit their interaction into the available representation system. In face-to-face meetings such as the one described above, participants often introduce additional representations — such as slides, handouts, scale models, drawings, or demos — to communicate information that would be difficult to present verbally. In an online interaction, the participants are generally unable to bring additional forms of media into the interaction, and so are forced to make use of the existing software system. This serves to exacerbate any mismatch between task information and representation system.

By analyzing the existing practice of a community of participants engaged in an ongoing collaboration, it is possible to improve a representation system that better matches the emergent complexities of the interaction. This requires a way to gather interaction data from a group of participants as they work, and effective methods for managing, searching, reviewing, and analyzing this data.

1.2 Recording interaction

Past work has focused on analyzing transcriptions of interaction, whether they be transcriptions of phone conversations (e.g., Schegloff 1979), videotape of participants (Suchman & Trigg 1991), or by other such investigative methods that allow recording of the users as they work. This work has some advantages over traditional ethnographic research: the observer effect, where participants change their behavior under observation, is reduced; the recording can be reviewed, allowing an analyst can examine an interaction over and over again, slow down areas of rapid activity, and go back to find the root causes of particular actions; and a single recording can easily be used by multiple analysts. Recording of ethnographic data by such means also allows the analyst to show the rcord to domain experts or the participants themselves, allowing for input from these sources during analysis. However, real-world audio and video are notoriously difficult media to index and search, even with recent improvements in skimming and summarizing technology. Because these media have no inherent structure or tagging, finding areas of interest and extracting quantitative conclusions from them is at best a time-consuming and work-intensive process; but for face-to-face communication, few alternatives exist.

For online collaboration, however, the experimenter's job is made easier by the very technology that enables the collaboration. Email, instant messaging systems, online chat, and web traffic all provide a certain degree of inherent structure. Past research has mined this available source of information, whether it is in the form of email headers and usage patterns (Whittaker & Sidner 1996) or instant messaging time/sender stamps (Nardi, Whittaker, and Bradner 2000), to provide quantitative data for ethnographic investigations of online collaboration. Additionally, by focusing on online interactions where the task information shared by participants is entirely mediated by computer, it is possible to record the entire interaction of the participants by logging the data transmitted by the computers.

1.3 New methods from old

We have devised methods for examining groupware systems based on an analysis of the interaction data that participants generate while performing collaborative work, with a focus on the discourse they create. Prior work (Clark & Wilkes-Gibbs 1986; Alterman & Garland 2001) has shown that groups with a greater degree of common ground require less communication to stay coordinated, and generate less communication in similar situations as their common ground increases. This indicates a strong tie between the discourse and the articulation work (Strauss 1985) performed by participants. An analyst can quantify and clarify the work that participants are doing by examining the quantity and types of communication that the participants engage in. We can then craft a representation system that promotes shared understanding and reduces work by matching the representation system to observed patterns of communication.

Communication can also reveal certain aspects of the cognitive load of participants. Our method examines the types and quantity of references that participants make, and tracks patterns of information flow. These observations reveal what information participants need access to as they perform their tasks. This allows an analyst to see what types of information participants coordinate over, how much information a participant must keep track of at any time, and how long they need to remember things, all indicators of cognitive load upon parfticipants. Additionally, the analyst can track how different types of information are used, what the paths of information flow from one participant to another are, and what representations are used to store each piece of information. These insights allow an analyst to suggest representation system designs that match, rather than conflict with, these patterns of information access.

While principled methods for good software design exist, and methods for generating and collecting usage data have been successfully adopted from psychology, methods for interpreting collected data are still fairly abstract. Existing methods provide an excellent understanding of the interaction but require a high degree of sophistication on the part of the analyst, can be difficult to teach, and may result in very different results when applied by different practitioners. These theoretical frameworks — including distributed cognition, activity theory, and cognitive work analysis — agree that it is necessary to observe the participants at work to fully understand the effort of a community of participants. However, these frameworks generally do not establish particular methods that an analyst can use to apply interpretations of these observations. These top-down methodologies primarily ask an analyst to interpret participant activities on a fairly abstract level, requiring a high level of sophistication from the analyst to extract design decisions.

Techniques for analyzing other forms of interaction do exist, and have been applied in a limited fashion to online coordination. Conversation analysis, in particular, provides techniques for examining transcribed conversation in a principled and thorough manner. These bottom-up techniques provide insight into the utterance-by-utterance communication of participants. However, these techniques require expansion in order to deal with long-term interaction, instead focusing on what the particulars of a conversation can say about the present interactions of the participants.

Our approach combines techniques borrowed from conversation analysis with insights from distributed cognition to create a methodology that:

1. can be applied to a wide variety of domains
2. can easily and quickly be taught to other analysts
3. produces similar results when applied by different analysts to the same interaction
4. provides a strong theoretical framework for looking at low-level empirical interaction data
5. produces quantitative data to support conclusions about the interaction
6. offers concrete design recommendations

To achieve these goals we developed two methods for analyzing interaction. The first examines the recurring problems that arise in the interaction as revealed by the secondary structure participants create in the discourse. Participants create conversational structure, such as formalized question/answer pairs, to manage difficult interactions. By examining this secondary structure, an analyst can hone in on problematic interactions. The second method provides insight into the frequency and importance of various topics of discussion by extracting the referential structure of the discourse; this enables the analyst to model information flow between participants and the cognitive load of individuals. For example, if participants are referring frequently to information of a particular type, the cognitive costs of accessing that information are magnified and must be kept small. Identifying frequently-used types of information allows an analyst to design custom representations that match the way participants access and share information.

It is important to emphasize that these discourse-level methods are meant to be used as a part of a larger work analysis and design methodology. They provide valuable insight into the minute-by-minute practices of workers, and the way that workers interact with the representation system available to them, but do not address important issues such as the social impact of work, worker mental or physical states, task efficiency, or the impact of personal interaction of workers. The next section embarks on a survey of relevant literature to provide historical perspective on the part that these methods can play in analyzing interaction.

2 Related work

A wide variety of approaches for analyzing work practice and designing new interactions have emerged over the years. Some are prescriptive, telling an analyst what patterns have succeeded or failed in the past. Some are analysis-based, and work backwards from the interaction toward redesigning the system. Whichever approach the methodologies take, they by and large structure an analyst's investigation of the work in some or all of a few basic ways:

1. They provide *rules for design* of an interaction. For example, usability guidelines encourage good design by heading off frequently-encountered problems and pointing out solutions to them.
2. They providing a *checklist of observations* to make. For example, Activity Theory encourages an analyst to identify the subject, artifacts, roles, community, goals, object, and so forth, as well as the interactions between them.
3. They provide a theoretical *framework for understanding* the interaction. For example, DCog encourages an analyst to look for the transference of information from representation to representation, and to understand the cognitive impact of the activities surrounding
4. They let the analyst *build a model* of the interaction. For example, GOMS provides a very structured way to describe specific activities, and provides tools to make use of the model thereby generated. In many cases (GOMS, workflow) the models produced can be used as inputs to programs which to simulate activity, with the potential for revealing unforeseen interactions.

The methodologies also each tend to center around a particular aspect of interaction, and build up a story about interaction based on that perspective. For example, Activity Theory focuses on the activity of the participants, that is, their interaction with each other and with the task is seen as a set of activities mediated by artifacts toward a certain object. This paper will present the methodologies grouped by their particular focus.

2.1 Prescriptive Frameworks

A variety of prescriptive and proscriptive guidelines have been produced over the years to share hard-won insights into system design: Schneiderman's basic rules for design

(Schneiderman 1998), Norman's principles for design of interfaces (Norman 1986b), patterns of UI design (e.g., Borchers 2001), and many others. These frameworks give designers a set of basic rules and abstractions that are important for good design of interactions. Designers can use these to construct systems that are more likely to be adopted by users. Analysts can also use them to understand why existing systems are not being successfully adopted, by noting situations where these guidelines have not been applied or are being improperly applied.

Such rules sets have been successfully applied to create easy-to-use, consistent application interfaces. The interfaces of many Macintosh applications have been strongly influenced by the original Human Interface Guidelines published by Apple Computer, Inc. (1987) for software developers. Modern website design, still a work in progress, has been strongly influenced by the work of Nielsen (2000), among many others. However, prescription-oriented frameworks tend to be quite abstract, are especially susceptible to being poorly implemented or applied too rigidly, and are only able to address interaction situations that have already been explored.

2.2 Task-based Analysis

Task-based analysis methods depend on a structural modeling of the tasks and goals of the participants. These methods make explicit the steps necessary to perform a task and the dependencies between different portions of a work process, and focus on how to best structure behavior to meet goals.

Time/Space Studies Research dating back to the beginning of the 20th century (Taylor 1911) examined the activity of workers in an attempt to make factories more efficient. By observing the work in practice that the laborers performed, Taylor identified ways to remove extraneous motions in tasks, introduced procedures that acted as coordinating artifacts to simplify the task, and identified places where division of labor could improve throughput.

For example, bricklayers of that time customarily spent a moment examining each brick before before placing it, so as to determine the best face to let show in the finished product. Taylor's recommendations included adding a worker whose sole job was to repack the bricks such that the best face of each brick was in the right orientation for the bricklayer to simply grab each brick and stick it in place. By introducing this relatively low-skill position to defray some of the work of the high-skill bricklayer positions, Taylor's method was able to improve the overall speed of production. Modern incarnations of this sort of process optimization are most visible in high-throughput arenas such as fast-food restaurants: motions required to cook, prepare, and serve food are closely analyzed to maximize the speed of production and minimize error rate, thereby increasing profits. Taylorism produces excellent results in efficiency improvements, but by and large ignores more complex issues such as interpersonal relations, coordination between workers, and in general the articulation work that is necessary to maintain group cohesion and to handle exceptions to the normal flow of interaction.

Task Analysis, GOMS, and Cognitive Work Analysis One descendent of time/space studies is Task Analysis (e.g., Kirwan & Ainsworth 1992). Task analysis breaks a worker's actions up into a set of tasks that must be fulfilled for a goal to be achieved, either organized linearly, or in a hierarchical tree of tasks and subtasks in hierarchical task analysis. These tasks often have interdependencies, preconditions, and knowledge requirements that must be fulfilled before the task can be accomplished. By mapping out these requirements, an analyst can make salient difficulties such as missing steps, mistimed steps, bottlenecks, and redundant work. From this a new design for the task can be constructed. Cognitive Task Analysis (e.g., Schraagen, Chipman, & Shalin 2000), which adds a personal-cognition aspect to task analysis, has also been applied to designing human computer interaction (Carey, Stammers, & Astley 1989). This method examines the mental load that a worker will be under at each stage of the task analysis, and takes into account the time and effort necessary to marshall, store and remember pertinent task details. GOMS (Card, Moran, & Newell 1983; John & Kieras 1996a) combines task analysis with second-by-second timing of computer interface actions. By incorporating empirically-derived timing for basic actions (such as mouse motion and key presses) and estimated timings for cognitive work (such as recalling items or making simple decisions), GOMS and its descendents allow an analyst to compute how long a typical, error-free operation will take to perform. This allows early pruning of poor design choices and rapid refinement of user interfaces, reducing development cost. GOMS has been used to aid design of domains ranging from text editors to nuclear power plant adjunct software (John & Kieras 1996b). GOMS, and its many descendents (some of which include more rigorous modeling of cognitive processes) are very useful for prototyping single-user interaction in situations where errors are infrequent and well-understood, but requires extension to fully address multi-user coordination and to delve into issues of information representation.

Cognitive Work Analysis Cognitive work analysis (Rasmussen 1986; Vicente 1999) extends the basic framework of task analysis into an analytic scheme that focuses heavily on the interactions between workers and their task. In addition to modeling the task domain and specific control tasks, as performed in task analysis, CWA models the strategies that workers perform to accomplish tasks, the impact of social organization and worker interactions, and the worker competencies required to perform these tasks. This added complexity of analysis situates CWA between pure task-based analysis and activity-based analysis methods. CWA has been applied to designs for a variety of domains, including construction of a book-lending system for a public library (ibid.) and analysis of medical domains.

2.3 Activity-based Analysis

Activity-based analysis methods focus on the activity of the participants, that is, the actual actions the people take when engaged in their work. These methods focus mainly on observation of participants in their place of work, and let an analyst note a variety of observations about their behavior. In contrast to task analysis, which concentrates on an idealized view of the steps by which the work should be done, activity-based analyses work backward, examining what constitutes the work practice from a higher level and how the work done derives the (potentially conflicting) motives of participants (Clancey 2002).

Workflow Analysis Workflow analysis focuses on the job, the product of the work participants perform. It examines the processes that affect that job and the impact and roles of the people as they relate to the job. It represents interaction as a complex flowchart with work nodes and decision points; jobs, representing an element of work, flow around the chart until they are completed. This model works well for piece-oriented environments such as a manufacturing facility. By visualizing the flow of work in this way, analysts can pick out bottlenecks, wasted loops of activity, and unnecessary steps in the interaction. They can then make recommendations about how to redesign the activity to reduce inefficiencies and speed production. It has been successfully applied to many corporate environments, and has found a home in the analysis of medical environments (e.g., Mueller et al. 1999). Due to its strong structural element, workflow analysis is well-suited to automation. There are a variety of commercial software packages available that allow a business to model its workflow and thereby draw conclusions. Some work has been done to construct a universal workflow vocabulary and identify workflow patterns (van der Aalst et al. 2003) that allow an analyst to easily chunk situations into manageable entities. In addition, workflow systems have been coupled with predicate logic systems to create simulation environments. One such system, Brahms (Sierhuis, Clancey & van Hoof 2003), allows an analyst to encode a work situation as a set of logical assertions and rules; this encoding serves as input to an agent simulator which generates activity based on the rules. This allows an analyst to thoroughly and inexpensively examine the implications of the designed behavior, potentially revealing problems that were difficult to notice in a static representation. Workflow is primarily focused on the product and process of work, whether that be material or conceptual, and as such leaves unexamined important considerations in online interaction design such as the specific representation of task information.

Activity Theory Activity Theory (Leont'ev 1978) addresses certain problems seen with task-based analysis by making social issues an explicit, first-class part of the analysis. By expanding the basic subject-tool-object mediation triangle (Vygotsky 1962) to include a cultural-historical perspective (e.g., Cole and Engeström 1993), researchers were able to formulate a perspective that included worker, task, and community in the scope of analysis. In this view, not only are there artifacts that mediate the worker's interaction with the task, but there are artifacts that mediate the worker's relation to the community at large. Bødker (1990) applied this framework to the design of interactive systems. She was able to recast the design process as a collaborative, social process. Elements of this sort of analysis were then used by others to, for example to help design physical interfaces for shared environments (Fjeld et al. 2002). Activity theory, as fairly high-level analysis method, examines conflicts within the interaction, and as such does not concern itself explicitly with examination of utterance-level features of the interaction.

2.4 Representation-based Analysis

Another class of analysis methods are primarily concerned with the representations of information that participants make use of during interaction: their form and purpose, the life cycle of information stored within them, and the procedures surrounding those representations for reading, storing, and transcribing information.

Distributed Cognition Distributed Cognition (DCog: Norman 1991; Hutchins 1995b; Holland, Hutchins, & Kirsh 2000) provides a theoretical framework for exploring the problem of introducing representations into a system of behavior. DCog views both participants and artifacts as important pieces of an interaction scenario. As in Vygotsky's view, artifacts act both modify the task and to affect the way participants reason about it. In this view, representations act as artifacts that are both external storage for task information and mediating tools that alter the nature of the task they assist.

The working system in DCog is a collection of representations, including a variety of external representations presented to an individual plus those internal to that individual. As the users juggle information between the various representations available to them they create procedures for ensuring smooth coordination. These procedures evolve into conventional solutions to recurring problems in transmitting, understanding, aligning, transcribing, storing, and retrieving information. From this perspective, the behavior of a system can be modeled as a combination of the set of representations of information, the procedures for propagating information between representations, and the actual flow of information between these representations.

In a collaborative scenario, artifacts can change the links between participants, fundamentally changing the way in which they interact. By altering the opportunities for interaction between participants, and the qualities of that contact, artifacts that mediate the collaboration can have a great impact on how efficient, effective, and pleasant that interaction is. Distributed Cognition offers a strong perspective for investigating the impact of introducing a new representation system into an ongoing practice. It has been applied to a variety of domains: the cockpit of an airliner (Hutchins 1995a; Hutchins & Klausen 1995), air traffic control (Halverson 1995), navigation of a ship (Hutchins 1995b), and so forth.

2.5 Discourse-based Analysis

Another set of techniques are more commonly performed post-activity from transcripts of the conversation (or more generally, the discourse). Discourse is generally easier to observe, record, and transcribe than the mental states of the participants, allowing these methods to be applied in a wide variety of domains with a minimal amount of invasive observation required.

Discourse analysis Discourse analysis (Stubbs 1983) is a method of understanding interaction which focuses on the discourse (spoken, textual, and otherwise) that participants create as a part of their interaction. Nuances such as word choice, sentence structure, the turn-taking of the participants, and other such features provide an analyst with insight into the function for the participants of the dialog and the problems that the participants may be having in achieving their goals. The intrinsically collaborative nature of dialogue, and the co-construction of meaning and intention that participants in a conversation necessarily engage in, mean that the principles of discourse analysis are also useful when analyzing collaborative activities. Discourse analysis has been put to good use in domains such as examining the differences in communication style of men and women (Tannen 1990).

Conversation Analysis Conversation analysis (Sacks, Schegloff, & Jefferson 1974; Schegloff 1991; Drew & Heritage 1992; etc.) also examines the conversation of participants in a joint activity. Conversation analysis grew out of ethnomethodology (Garfinkel 1967), a way of examining how participants in a social situation cooperate to make sense of their interaction. Conversation analysis examines interaction by focusing on the dialogue that participants generate as they perform their tasks: the turns that speakers take, the construction of utterances for the benefit of listeners, and the way that information is passed between participants. Focusing on conversation itself allows an analyst to use this easily-accessible data to investigate issues that are otherwise hard to observe in an ongoing interaction.

Conversation analysis has been put to use examining a wide variety of situations. Early work looked at telephone conversations (e.g., Schegloff & Sacks 1973); more recently, it has also been applied to a variety of other domains, such as examining the conversations of doctors with their psychotic patients (McCabe et al. 2001).

3 Techniques

This paper will now present an in-depth explanation of the two analysis techniques we have created. This will be followed by experimental evidence for the utility of these techniques, and discuss the results of experiments that test the efficacy of both of these methods in a handful of domains. It will then conclude with discussion of an experimental test of how the methods were taught to and used by a mixed class of graduate and undergraduate students.

- **Recurrence analysis** concentrates on social issues by examining recurring interactions, errors, and the secondary discourse structure participants create to handle problems.
- **Referential structure analysis** concentrates on issues of topic and information flow by examining the referential structure of the participants' discourse; these data have significance in measuring individual cognitive load.

These methods arose from application of elements of both conversation analysis and distributed cognition to analysis of groupware systems. A brief discussion of the evolution of the methods follows.

3.1 From conversation analysis to interaction design

Sacks, Schegloff, and Jefferson (1974) assert that conversation is a party-administered, locally-managed behavior. That is, conversation is organized by a set of local conventions to determine next speaker, transition between speakers, negotiate the ending of a conversation, and for other such recurring problems of coordination (Schegloff and Sacks 1973). In addition, participants in a recurring activity habitually create and participate in conventionalized plans for behavior in stereotypical situations; these plans simplify interaction by creating expectations in other participants.

Coordinative structures such as these function as conventions (Lewis 1969; Clark 1996), community-specific solutions to recurring problems participants have organizing their conversation and activity. For example, problems such as figuring out who gets to speak next, what to do when meeting a person, or other such common situations, can be resolved by adhering to a convention for behavior. These conventions provides mutual expectations for behavior on the part of the participants. While the individual, internal representations of these conventions can never be identical, the mutual expectations for behavior and meaning that they create serve to reduce the effort required to interact. For example, two people who share a common societal convention of introducing oneself to a new acquaintance will find it much easier to communicate than those who must create such activity extemporaneously. Our research group has shown quantitatively that the coordinative and communicative effort required to perform a collaborative task is reduced by the introduction of conventions for conversation and action (Alterman et al. 2001). By organizing task behavior and providing expectations about the behavior of others, these conventions form a strong basis of common ground, and reduce the articulation work necessary to perform a task.

One way of realizing conventions for behavior is to generate *secondary structure* in the discourse which serves to organize behavior. In an ongoing collaboration, participants faced with a difficult coordinative problem will often attempt to generate this secondary structure in their interaction to address the problem. Secondary structure provides organization for their talk about the task at hand, which in turn can organize the task itself. One example of secondary structure can be seen in a canonical opening for an impromptu meeting: "I have a few questions for you. First, . . . " This preliminary, straight-forward organization creates expectations about the roles of the speaker and listener and provides a shared plan for the rest of the interaction (Schegloff 1980). Faced with more complex coordination problems, participants often generate more complicated structures to simplify their coordination. In a household, a centralized grocery list, with the attendant procedures for maintaining the accuracy and consistency of that list, provides helpful structure to simplify coordination of shopping for groceries.

If this impromptu structure and the procedures surrounding it proves successful — that is, if they serve to reduce the work required to successfully complete the task — then it can be advantageous to solidify it into concrete conventions or in fixed form as a coordinative artifact. This serves both to make the artifact perceivable and available to all current participants, and to benefit future participants in the interaction or in similar interactions. However, realizing artifacts may prove quite difficult in practice, because the form of the interaction is necessarily changed by the introduction of a new cognitive artifact. The effects that this has on that interaction are difficult to foresee. Often coordinative artifacts are introduced with good intentions but end up making things worse.

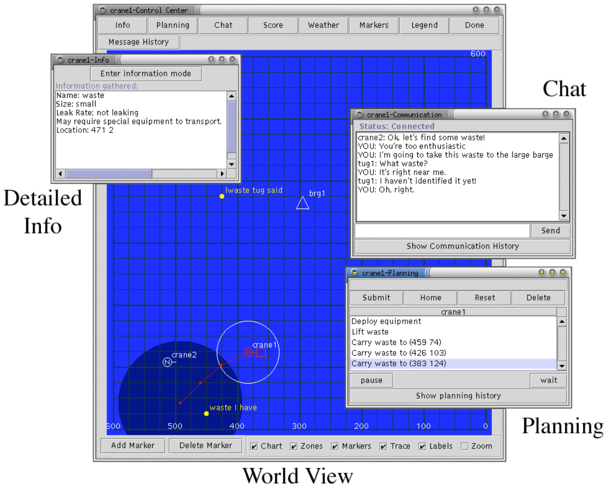
Analyzing the impact of alterations in the representation system requires a theoretical framework within which to examine the interaction of participants and artifacts. While the structure of internal representations is a matter of some debate, the structure of external representations is available for analysis. The operations performed by participants on external representations can be directly examined in the attempt to understand how the participants are interacting with the information available to them. This includes their interactions with each other as mediated by the shared workspace. There are some issues with this approach — for example, it is clear that different users, and different groups of users, interact slightly differently with the same representations. Focusing on the actual use of a representation system, then, is crucial to understanding how it is being used in practice, which requires analyzing coordination work directly.

3.2 Analyzing coordination work

Collecting complete and contextualized ethnographic data of a cognitive system "at work" enables an analyst to draw strong conclusions about the cognitive and social effort imposed on the participants. However, as noted earlier, collecting such data is difficult and time-consuming. There are significant problems in reviewing, summarizing, analyzing, and drawing conclusions from the enormous volume of data created by real-time collaboration. Ideally, an ongoing interaction would be captured in a fashion that allows detailed but rapid review without compromising accuracy. To address these issues, our research group has built a component-based groupware toolkit, THYME (Landsman and Alterman 2003), to allow a system designer to quickly and easily generate groupware systems that will automatically record interaction data as a collaboration progresses. The data can then be replayed using SAGE, a data playback program (Landsman and Alterman 2002), and analyzed with a variety of tools for performing both qualitative and quantitative analyses of the data.

One of the systems we created to study issues in same-time/different-place coordination is the VesselWorld groupware system. VesselWorld is a turn-based multi-user simulation where three users situated at separate computers conduct a clean up of a harbor via a graphical interface. Though the users cannot see or hear each other, they are able to chat via the VesselWorld interface. The simulated harbor contains toxic waste that must be safely retrieved and loaded onto a large waste barge. As the users interact, the system logs all actions and communication for later analysis. The ability to generate and play back transcripts of interaction makes VesselWorld ideal for exploring issues of group interaction. At this time, we have collected over 300 hours of data from more than 20 groups using various versions of VesselWorld.

In the VesselWorld system, each user acts as the captain of a ship navigating the harbor. Two users pilot ships with waste-retrieval cranes attached (referred to as crane1 and crane2), allowing them to lift and load barrels of toxic waste; the other user pilots a tugboat (referred to as tug1), and is able to move small barges around the harbor, identify waste, and seal the leaks caused by mishandling of waste. Each user is only able to see a small nearby region of the harbor. The harbor is cleared in a turn-based fashion, with each user explicitly planning an action for a turn before submitting them to the system for



Name: waste
Size: small
Leak Rate: not leaking
May require special equipment to transport.
Location: 471 2

crane2: Ok, let's find some waste!
YOU: You're too enthusiastic
YOU: I'm going to take this waste to the large barge
tug1: What waste?
YOU: It's right near me.
tug1: I haven't identified it yet!
YOU: Oh, right.

Deploy equipment
Lift waste
Carry waste to (459 74)
Carry waste to (426 103)
Carry waste to (383 124)

Figure 2: The VesselWorld interface

evaluation. During a session, the users are physically separated, but are able to communicate freely via a textual chat facility built into the VesselWorld system. The basic interface is shown in Figure 2. The large central window shows the harbor, with the small portion of the harbor currently visible to the user shown as a darker circle in the lower left. Clockwise from the left, the smaller windows are: the Info window, displaying detailed information about objects in the harbor; the Chat window, allowing textual communication with other users; and the Planning window, containing the user's current plan for domain actions and the controls for editing or submitting that plan.

For each experiment, the group of three participants is trained in use of the system and then asked to solve a series of waste retrieval scenarios, each generally requiring one or two hours to complete. While there is no time constraint imposed, emphasis is placed on minimizing the number of turns of action required and the number of waste handling errors. A group score provides feedback to the users as to their progress. This proved an adequate tool to foster involvement in finding an efficient solution.

VesselWorld participants have a small set of representations available to them for coordinating their activity. Primary among these is the textual chat, which provides a very flexible means of expression. In addition, there is evidence of participants coordinating their activity by taking advantage of the visibility of actions of nearby vessels in the harbor: when two vessels are close enough to actually see each other's current state, participants make use of this information in their planning. Participants also made extensive use of private markers, which provide a way to place an annotation (visible only to the user) on a section of the harbor. Users used private markers extensively to keep track of waste information.

We performed a discourse analysis of the participant dialog from a VesselWorld pilot study. We found three main indicators which suggest that incorporating an alternate representation might reduce participant effort:

1. recurrent patterns of coordination
2. recurrent errors in coordination
3. creation of secondary structure to coordinate activities

Each of these indicate that the representation system provided for the participants may not match well with the way the participants share and use information. This section will examine each of these indicators in order, and illustrate with examples how their occurrence reflects incompatibility between the available representation and the necessary coordination of the participants. The examples are taken from an experimental study, the "VW3" VesselWorld experiment, which will be explored further later in the paper.

3.2.1 Recurrent patterns of coordination

In situations characterized by the first indicator, participants find themselves repeatedly in a particular sort of situation. Because of the sheer volume of interaction in this situation, small improvements in coordinative efficiency will have large returns. This is similar to the case of a software engineer who improves the efficiency of a critical loop of code that is called very frequently even small performance gains in such a pivotal place are significant when multiplied by a large number of iterations. (Our second technique, explained in the next section, allows us to quantify this.)

Analysis of the VesselWorld data revealed a number of such routines. One of the most frequent patterns involved the reporting of waste information. A large portion of the communication participants generate during the early part of the session consists of participants reporting the discovery of new barrels of waste. Due to the nature of the task, waste could be discovered by any user, but each waste required a particular set of actions involving one, two, or all three users to handle successfully. For this reason, succesful clearing of the harbor depended on participants sharing information about newly-discovered wastes. Because of the frequent reporting, each group eventually settled on their own stylized vocabulary and interaction pattern for reporting wastes. Despite the conventions that groups created for reporting waste information, the task of not only reporting but also understanding, transcribing, and remembering the information was cumbersome and error-prone.

3.2.2 Recurrent errors in coordination

The second indicator, recurrent errors, points to areas where participants are having acute difficulty in maintaining coordination. For this case the analyst should pay special attention to incidents where participants fail in joint activity due to misalignment of expectations or perceptions. Note that the difference between this indicator and the previous one is somewhat subtle; here, the situation may not be as frequent, but the coordination required is so error-prone that the participants frequently fail to perform their joint activity successfully.

We found recurrent errors in situations such as recall of waste information, planning of future actions, and planning and execution of joint actions. Understanding and recall of waste information was especially problematic. Discrepencies frequently intruded into the flow of information, creeping into each step involved in discovering new waste: reporting waste information, understanding that report, properly transcribing it to a local representation (whether internal or external), and recalling it from that representation when the time came to act. These errors could be quite pernicious, as an error in recall would not be apparent until participants went to act on the erroneous knowledge. One such situation, typical of errors seen in all groups, is shown in Figure 3.

crane2: what eq is needed for the small on top of the attached barge
crane1: none
tug1: Dredge
crane1: huh? i thought that was the sm none?
tug1: It apparently isn't.
crane1: k

Figure 3: Mistakes in recalling waste information lead to confusion

In this transcript, crane1 did not simply misremember the pertinent waste information (that equipment required was "none"), but had transcribed it improperly into his local representation: he had created a private marker with erroneous information. The tug, source of authoritative information about equipment requirements, also responds, revealing a discrepancy. The participants then need to engage in a brief repair to ascertain which version of the information is accurate. In this case, because the mismatch was caught before action was taken, no dropping of waste or leaks occurred, but incorrect information caused a significant increase in required teamwork.

3.2.3 Creation of secondary structure

The last indicator goes a step further than the first two; in this case, the participants have both determined a potential area of difficulty and have devised structure to improve the situation. However, this structure may not be sufficient to eliminate coordination difficulty completely; in most cases where we saw such structure evolve, it was at best a cumbersome measure to attempt to reduce the number of errors. This was primarily because the tools provided to the users were not adequate to provide seamless solutions; in most cases, the structure generated consisted of ritualized sets of conversation that provided a procedure the participants followed to perform certain recurrent tasks. Participants were unable to generate coordinative structure which fully addressed the difficulties they were attempting to mitigate. Nevertheless, the structures they create are revealing.

In joint lifts, where the two cranes needed to coordinate their domain actions to lift a large or extra-large waste, timing of the joint actions was very error-prone. Users were not able to see directly when plans had been submitted to the system; this lead to problems where ambiguous statements such as “submit a lift next step” cause confusion about the current state of the joint operation. This caused many mistakes and a great deal of frustration for the users. In some groups, the participants eventually established structural conventions in their discourse to organize their actions. An example of the sort of secondary structure created by participants can be seen in Figure 4. The two cranes must conduct a joint lift of a large waste by submitting the same plan at the same time. Lack of visibility of other users’ planned actions created difficult timing problems. After only a few repetitions, structure such as these adjacency pairs (Schegloff & Sacks 1973) appeared.

crane1: sub Lift
crane2: k
crane1: sub Load
crane2: k

Figure 4: Adjacency pairs in VesselWorld dialog

Here, crane1 is proposing and confirming each step in the shared plan: first, to submit a step to jointly lift a previously discussed waste ("sub Lift"), and then to submit a step to load it on a waste barge ("sub Load"). In each case crane2 explicitly acknowledges both the plan and the timing; crane1 irrevocably commits to his plan only after receiving the acknowledgement from the other participant. This structure ensures that the two cranes have matching plans, and are maintaining coherence of expectations, hence resolving the issues with timing.

Another example of secondary structure involved the Marker Check (shown in Figure 5), a complex procedure invented by one group to attempt to align the private representations the users had for waste information. As in previous examples, one user's private representation is in error, but it is not clear which representation is correct. By reviewing the contents of one user's private representation, and having each user compare that to their private representation, the group was able to successfully align their individual representations.

crane1: [ALL] MArker CHECK: You should have 13 (thirteen) WASTE MARKERS. Confirm
⋮
crane1: Legend: (Sm—L—XL)-(Ni (not id'd) Net — Dr)
crane1: From south east clockwise
crane1: (Sm-NI 50,0) (Sm-NET 150,25) (Sm-NI 350,150) (Sm-NI 550,50) (Sm-NI 600,100) (thats all south of equator. NORTH coming up
tug1: 97,441 and 72,368 already ID'd
crane2: 350,150 is barge, isn't it?
crane2: that's the problem

Figure 5: Marker check reveals a discrepancy in the users' private representations

However, producing and using this structure proved quite time-consuming, and the procedure was itself error-prone. Because of the limited tools available to the participants to structure their work they were not always able to successfully construct solutions. There is no guarantee that the organizational structure that the users add will improve the situation at all; it is possible that some problems of coordination are best dealt with using a context-free form of communication like textual chatting. In general, however, introducing structured representations seems to improve such situations. In the next section, we present experimental evidence that introducing well-chosen alternative representations significantly improves performance.

3.2.4 Applying recurrence analysis to experimental data

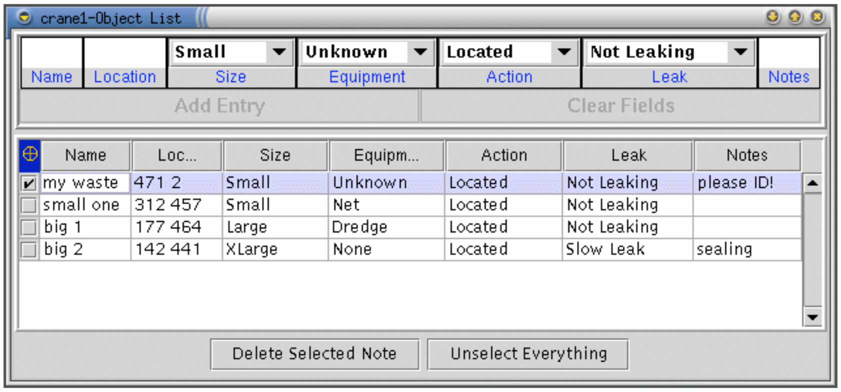
Analysis of a pilot study of VesselWorld (generating about 60 hours of interaction data) revealed three recurring areas of difficulty in coordination:

1. *Shared domain object* naming, reference, and information sharing.
2. *Timing of activities* consisting of closely coupled cooperative actions.
3. *Higher-level planning* to manage multiple cooperative activities in searching the harbor and organizing the removal of all the wastes.

Each area was associated with a noticeable increase in the amount of coordination work required for participants to complete their task, and were the source of the majority of errors that participants committed. To address these difficulty we redesigned the representation system available to users of VesselWorld. The redesigned version included three new representations created to ameliorate the coordination problems participants were encountering.

One of the new coordinating representations, the Object List, is shown in Figure 6. It was constructed to resolve recurring errors in naming, sharing, and recalling information about shared domain objects such as barrels of toxic waste. The central section of the window is a list of notes about the toxic wastes that have been reported by users. Each row represents a single waste. This list is visible to all users; all users can edit any entry, add new entries using the palette at the top of the window, and delete any waste entry. The columns of information were selected by examining what waste features users talked about the most: a way to associate a *name* with a waste; its *size*, *location*, and necessary *equipment*; the current *action* required on the waste; and whether or not the waste was *leaking*. Based on indications that semi-structured representations are in general more useful than ones that force users to cast information wholly into a fixed representation (Malone et al., 1987) we added a free-form notes area for users to note information about a waste that did not fit into the structure presented.

We then conducted a single-variable experiment (the VW3 experiment) to assess the impact of these three coordinating representations on the performance of groups of subjects using VesselWorld. One set of groups, the control (which we will call the non-CR groups),



Field	Name	Location	Size	Equipment	Action	Leak	Notes
			Small	Unknown	Located	Not Leaking	

	Name	Loc...	Size	Equipm...	Action	Leak	Notes
☑	my waste	471 2	Small	Unknown	Located	Not Leaking	please ID!
☐	small one	312 457	Small	Net	Located	Not Leaking	
☐	big 1	177 464	Large	Dredge	Located	Not Leaking	
☐	big 2	142 441	XLarge	None	Located	Slow Leak	sealing

Figure 6: A coordinating representation: the Object List

used a version of VesselWorld similar to the one in the pilot study but with improved stability. The other set of groups (the CR groups) used a version of VesselWorld with the coordinating representations enabled.

Each set consisted of three groups of three subjects. The groups were made up of a mix of area professionals and undergraduate students; all were paid a flat fee for the experiment. Each group was trained together for two hours in use of their system, and then solved VesselWorld problems for approximately ten hours. To alleviate fatigue concerns, the experiment was split into three four-hour sessions. Subjects were asked to fill out entrance surveys to obtain population data and exit surveys to get feedback about their experience with the system and coordination issues arising in their group.

A set of random problems was produced, and subjects were given a succession of problems drawn from this set. However, groups did not necessarily see the same problems, nor in the same order, and because of differences in performance, did not complete the same number of problems over their ten hours of problem solving. To account for this, a general measure of the complexity of a particular problem was devised, taking into account the quantity and type of the wastes in the harbor, their distance from the large barge, and the number of small barges available to the subjects. This metric was used to normalize results. The results presented are a comparison of the final five hours of play for each group, by which point the performance of each group had stabilized.

Expected results The experiment produced a number of major results, summarized in Figure 7. The performance of the CR groups was significantly better than non-CR groups according to many measures: clock time necessary to solve a problem, interface work (measured as the number of system events generated per minute), and number of errors committed that resulted in waste leaks. Performance in some of the trouble areas we had previously identified — close coordination, domain object reference — was notably improved, with errors due to miscommunication of object information significantly reduced.

Measure	Non-CR groups	CR groups	Improvement
Communication (lines per minute)	5.53	2.35	58% ($p < 0.01$)
Solution time (minutes per session)	96.7	56.6	52% ($p < 0.01$)
Interface work (system events per minute)	886	514	42% ($p < 0.05$)
Speed of play (rounds per minute)	1.29	1.73	34% ($p < 0.05$)
Mistakes (errors per minute)	0.121	0.047	62% ($p < 0.2$)
Efficiency (rounds per complexity)	1.51	1.09	28% ($p < 0.35$)

Figure 7: Comparison of CR and non-CR groups in VesselWorld

The most significant effect, though not the one of greatest magnitude, is the 58% reduction in communication generated per minute. Also highly significant is the 42% reduction in clock time per session. Only slightly less significant is the reduction in system events (mouse clicks, etc.), down 52%. Coupled with the result for the increase in rounds of activity per minute — up 34% — we see that the CR groups worked faster with less interface effort. These results were all expected; the alternate representation system provided allowed users to work faster and with less communication necessary in the chat window.

Also as expected, overall domain errors (errors in performing domain actions which led to a toxic spill) were reduced by 62%, but variance of this measure was quite high due to the low frequency of errors; this reduced its confidence below statistical significance ($p < 0.2$). One measure that we expected to drop significantly was the number of rounds of activity required to perform the task. However, as can be seen below, while the reduction in this measure was promising, it was found to be not statistically significant.

Unexpected results Because of the relatively small sample population variability of group performance due to individual differences was high. Real-world issues interfered with data collection; for example, personal strife between subjects in one group led to severely reduced performance in early sessions. Likewise, one subject's comparatively low computer proficiency introduced a bias in that group's clock time. But other than decreasing confidence in statistical results these outliers were not problematic.

A more troublesome result was that certain columns of the Object List went unused. Specifically, the Action column, meant to aid users in tracking the next action to be performed on a waste, and the Leaking column, used to indicate a waste was leaking, went almost entirely unused. In the exit survey, one user wrote: "...the Object List had too many options. Many weren't used because we were in constant chat contact."

Most disconcertingly, the high-level Strategy window was not used at all once training in it was complete. Subjects in the CR groups gave some insight into why: "We never used the Strategy window because we could see what we were doing in the planning window." That is, the users felt the representation did not match the way they handled the information it was supposed to be storing. One user's assessment of the fundamental problems with the High-level Planning CR was especially interesting: "...since all plans must de facto be agreed upon by all (relevant) players, negotiation via the Chat window is required. Since the plans are discussed in detail there, putting those plans in the Strategy window would be redundant." This sort of social impact on the way users handle information was completely unanticipated.

Our analysis method left us with no explanations as to why the "Leak" and "Action" columns of the Object List, and the Strategy window itself, went unused by all groups while others were used constantly. We were unable to anticipate these failures of design using our existing analysis methodology. Because of this we saw the need for a more detailed form of analysis, leading to the creation of the referential structure analysis techniques.

3.3 Analyzing referential structure

After reviewing the data, we noted that many of the issues we saw with the existing representation system could be attributed to a mismatch between how the representations mediated interaction with information and how that information was used by participants. To examine these issues, we needed a method that would highlight how information was used by the participants. To this end, we examined the life cycle of information as it flowed from representation to representation within the system. Other methods, including workflow analysis and distributed cognition, address these issues; in this case, we examine the motion of specific pieces of information by noting references to it in the discourse.

In VesselWorld, information usually has a simple life cycle. For example, information about domain objects is first discovered via exploration; optionally, the discoverer uses the Info Window to retrieve ancillary details; the information is then reported, either in the chat window, or (in the CR groups) via the Object List or Shared Planning window; it may then be noted by other users; at some future time, it again becomes relevant and must be retrieved; and finally the waste is dealt with and the information becomes irrelevant. Each of these operations on the information represent access to that information; together, these accesses form the information access pattern for that piece of data.

Information that is relevant over a long span of time may be updated, modified, or otherwise accessed before it is finally rendered irrelevant and forgotten. While information may reside briefly in the short-term memory of the users, storing complex information there can put an unreasonable cognitive burden on the user; distributing the information into the environment can yield superior results. However, presenting too much information can lead to information overload. It is important, therefore, to determine what information is worth mediating with an external representation system. We formed two hypotheses:

1. Information with a long period of relevance is worth recording in an external representation
2. Information which is accessed frequently by the users is worth recording in an external representation

These hypotheses serve as general indicators of what sorts of information the representation system should store to simplify coordination. In the first case, information that is relevant over a long period is very likely to be irrelevant for some subsection of that period. During this time, it represents an unnecessary burden on the user's short-term memory, and in a complex situation with many such items, the burden can easily outstrip the user's ability to memorize. Therefore, the ability to shift that information to a readily accessible external representation is an opportunity to reduce the cognitive load of the participants. In the second situation, the information is used frequently, and hence needs to be readily available to the users. However, there may be too much information for the user to keep it all in short-term memory. Therefore the information needs to be easily and continuously accessible to users as they perform their tasks, and again, off-loading it into an external representation should decrease effort.

Iotas In order to highlight the types of information that exhibit the characteristics outlined above, an analyst must track the referents users refer to by thorough examination of the discourse. We developed a system for tracking the references that participants make in sharing information. From this we can examine what information is communicated, when, and in what fashion. Our scheme focuses on identifying the information that users share and on following its subsequent use within the system.

To examine the referential structure of the discourse, the analyst must examine the dialog produced by the interaction line by line. On each line, the participant communicating may make one or more references. The goal here is to track these references and either identify them as new information that has not been shared before, or connect them to previously shared referents. To avoid the tongue-twister "references to referents", and the overloaded word "reference", we call the referent that a set of references point to "iotas" and talk about mentions of an iota. An iota, for our purposes, represents a simple conversational item that the users refer to. Examples include: a barrel of toxic waste in the VesselWorld domain; a plan to clear a barrel of waste; a realignment of discrepant personal representations; or a conventional procedure for handling a particular situation. In general, any sort of information that the users refer to qualifies as an iota. However, it is important to note that because references in conversation may refer to an object that does not exist, the iota may represent a ficticious referent that exists only in the mind of some of the participants, and that multiple iotas may exist for a single "real" referent, because different participants refer to it as entirely separate objects. Rather than represent some authoritative set of objects, iotas represent objects that any participant refers to.

Iota types and tokens To allow investigation of the differences between various sorts of information, we assign each iota a type. These types are a combination of domain-specific types — in VesselWorld, these include wastes, vessels, locations, and barges — and domain-independent types, including plans and repairs. Plans are references to discussion of future action, of plans in progress, or plans that have been completed and need discussion. Repairs are iotas where users attempted to fix mismatches in common ground, correct errors in the interaction, or disambiguate misunderstandings. As in Lockman & Klappholz (1978), every referential object in the discourse is an iota that can be tracked by examining the referential structure.

Given a set of collaborative tasks, the hypothesis embedded in the method is that iota types will reflect the structure that participants will use to share information and organize their activity. There is a type/token distinction here: the types of iotas identified tell the analyst the sort of topics that participants discuss. The specific instances of an iota, treated as tokens, can be used to form conclusions about how each piece of information is handled by participants.

In reality, the observation and the definition are intertwined; two iotas types that are handled the same way may be better represented as the same type, whereas a class of information whose use can be split into two or more distinct usage patterns may need to be reclassified as being made up of two or more iota types. As analysis progresses the analyst will generally have to iteratively reassess the choice of iota types until an acceptable set of types is derived for the specific goals of the analysis.

Iota typing can also be supplemented by subtypes or aspects. For example, the waste iotas found were talked about differently according to what aspect of the waste was under discussion. A reference to the equipment needed for a waste might be handled differently than a reference to the size of the waste. The particular scheme for identifying iota types is at the discretion of the analyst. However, identifying iota types is not just an idle exercise. As will be shown, the process of identifying an appropriate set of iota types provides significant insight into the sorts of information that participants exchange.

3.3.1 A sample referential structure analysis

Once iota types are identified the job of the analyst is to go through the transcript and mark up each reference to an iota. To clarify the methodology, let us follow through an example taken from a non-CR group in their final VesselWorld session.

7.	tug1:	mX at 400 125	[Iota–7a waste: mx@400,125]
8.	crane1:	medium at 392 127	[Iota–8a waste: m?@392,127]
9.	crane1:	that's got to be the same one	[Iota–9a repair: Iota–8a is Iota–7a]
10.	tug1:	yep	Iota–9a
11.	tug1:	that's an mX	[Iota–7a waste: mx@392,127]

Figure 8: Applying referential structure analysis

In Figure 8, three VesselWorld subjects have encountered a few wastes, and are sharing what they see so they may plan how to clean up the harbor. First, in line 7, the user operating the tug vessel reports information about a nearby waste, using an established shorthand: "mX at 400 125". Here, mX indicates that the tug is talking about a waste of medium size (m) that requires no special equipment (X) to handle safely. The tug indicates that the waste can be found at the location (400, 125) in the harbor. To create a waste iota for this referent, the analyst generates a unique name based on the current line of discourse (IOTA–7A), puts the iota in square brackets to indicate that this is a new or modified reference, denotes the type of the iota as "waste", and lists all information available about that waste.

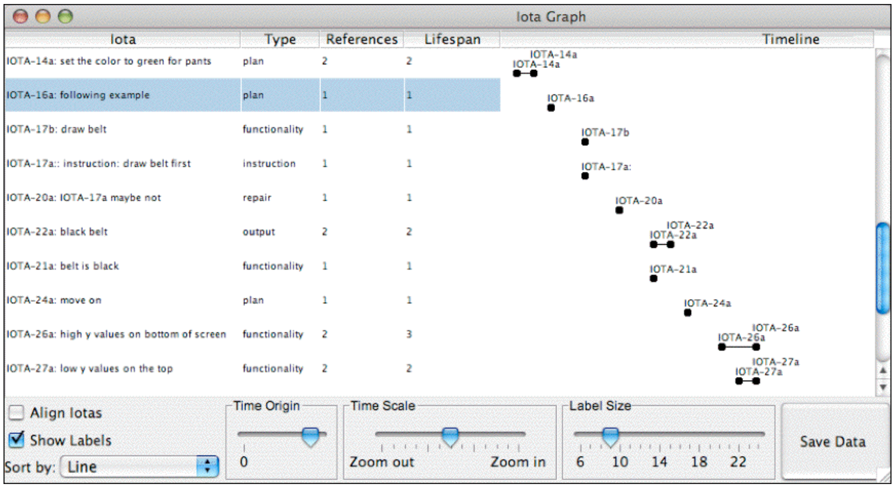
On line 8, crane1 simultaneously reports on a waste nearby. The analyst again creates an iota to track it. Here, the type of equipment needed is unknown to the user, as only the tug can ascertain equipment needs. The user indicates this by omission; the analyst instead uses a question mark in the iota definition. In line 9, crane1 notes the similarities between the two waste reports: both wastes are medium-sized, and they are located very close to each other. The reports of equipment (unknown vs. none) are not contradictory. Also, crane1 can likely see the area the tug is referring to, and does not see a waste there. The users have run into this situation before, and so crane1 quickly proposes (in line 9) a need for repair of common ground. The analyst notes the repair as an iota of type "repair", and makes sure to mention the iotas that are involved.

The tug, who can also see both locations (400,125 and 392,127), and is able to refer to the Info Window to get the exact coordinates for the waste, implicitly agrees to (and therefore refers to) the repair in line 10. It appears that the tug estimated the original specification of the waste location (400,125), rounding to the grid intervals visible on the user's display. The analyst notes the agreement to the repair, and refers to the already-instantiated IOTA–9A by naming it without square brackets. In line 11, the tug reviews relevant information about the waste. This acts as evidence supporting the repair (that the two references refer to the same waste). The analyst updates the expansion of IOTA–7A (again using square brackets, this time to indicate that the contents of the iota are being modified), and chooses the earlier of the two names for the waste (IOTA–7A and IOTA–8A) to disambiguate further references to the waste.

In this brief example, we have identified two types of iotas: waste iotas and repair iotas. There are three tokens: IOTA–7A, a waste iota; IOTA–8A, another waste iota; and IOTA–9A, a repair iota.

3.3.2 Interpreting data

The tagging of the dialogue yields a set of and analysis of data can be performed using Lyze, a software tool that aids the analyst in marking up a transcript and visualizing the results. Lyze allows the analyst to quickly and easily compile an analysis to extract information from tagging performed. It automatically summarizes the iota references entered in a timeline graph, a graphical representation of the iota data, as shown in Figure 9. Each line represents the life of a single iota token, with dots along the line representing each reference to that iota. By zooming out you can get an overview of the general trends of access; by zooming in you can see the particulars of how each iota is handled via the references made to it. Sorting by iota type can reveal similarities among iotas; sorting other measures, such as lifetime, reveals similarities or disparities across iota categories.



Iota	Type	References	Lifespan
IOTA-14a: set the color to green for pants	plan	2	2
IOTA-16a: following example	plan	1	1
IOTA-17b: draw belt	functionality	1	1
IOTA-17a:: instruction: draw belt first	instruction	1	1
IOTA-20a: IOTA-17a maybe not	repair	1	1
IOTA-22a: black belt	output	2	2
IOTA-21a: belt is black	functionality	1	1
IOTA-24a: move on	plan	1	1
IOTA-26a: high y values on bottom of screen	functionality	2	3
IOTA-27a: low y values on the top	functionality	2	2

Figure 9: The Lyze tool graphically displays iota life cycles

The tool is constructed to run alongside a playback tool such as SAGE, which allows the analyst to examine the interaction carefully to attempt to understand the discourse. Coupling the visualization of referential structure with the ability to review the interaction allows further refinement of the analysis.

The tool automatically computes a number of statistics for the iota data. Among the statistics the program outputs are:

- *frequency of appearance* of each type of iota, out of the pool of all iotas.
- *number of mentions* of each iota token.

- *lifetime* of each iota token, defined as the number of utterances between the first and last mentions.
- *density* of each iota token, defined as the ratio of mentions to lifetime.

Other statistics could also be drawn from the base iota data. These basic statistics give the analyst some insight into how information is accessed by participants. The data can be exported in a common data format to allow import into spreadsheet or statistical analysis programs, allowing application of the powerful visualization tools provided by such programs.

3.3.3 Exploring and visualizing experimental data

The referential structure analysis can be used draw conclusions regarding how to redesign a representation system based on observation of its use. Fundamentally, this includes the observation that (other things being equal) the smaller the quantity of task-related chat, the more fluid and error-free the interaction is. In addition, the quantity of communication about objects gives indication as to the magnitude of the mismatch in common ground — the more common ground shared, the less participants need to communicate. As the referential analysis method provides quantitative numbers for the amount of task-related dialog that occurs, it can be used to indicate situations where the participants are having difficulty maintaining alignment of their personal representations of shared information.
In addition, because the method reveals precisely what topics the participants talk the most about, it provides insight into exactly what portions of common ground are providing the difficulty. This lets the analyst direct redesign efforts appropriately. Finally, because the technique can be used to investigate a redesign of the system, it can be used to directly verify the utility of a redesigned representation system.

Experimental results We applied referential structure analysis to a number of VesselWorld log files in an attempt to further explore and understand the interaction and examine the flaws in our previous design of a representation system. A summary of the data for the non-CR groups appears in Figure 10.

This analysis yielded some intriguing results. Most notable was the obvious differences between plans and wastes — the two most common types of iotas seen. Plan iotas, by far the most numerous type of iota, tended to have a short lifetime (averaging 12 lines of chat). In comparison, wastes were relevant for a much longer period (averaging about 169

Iota Type	Frequency	Mentions	Lifetime	Density
Plan	57%	3.4	12.0	28.5%
Waste	17%	6.6	168.7	3.9%
Location	8%	2.6	62.6	4.2%
Repair	8%	3.0	4.8	62.5%
Barge	4%	11.9	294.0	5.6%
Vessel	4%	3.1	183.6	1.7%

Figure 10: Referential structure data from the VW3 experiment

lines of chat). This meant that information about a waste had to be retained in some representation for that rather long period — either in a participant's memory or in one of the available external representations. This indicates that there is an additional cognitive load incurred in having the system of interface plus user remember the information, which could be expressed as the cost of memorizing and later remembering the waste information, the cost of transcribing that information to an alternate representation such as markers, or perhaps as the cost to later reacquire that information from environment. This sort of transcription is necessary due to the quantity and complexity of the waste information; participants are simply unable to store the relevant information in working memory. In any case, it is an indication that providing a way to easily transcribe this information will reduce the cognitive load.

Another obvious result was the difference in density between plans, wastes, and repairs. Density is a rough measure of how dominant the topic represented by an iota is in the conversation. An iota with a high density can be referred to in the majority of all utterances over its lifetime. Repairs (with an average density of 63%) did just this, completely dominating conversation when they occurred. As a result, it seems unlikely that they would require a new external representation to mediate; because of their tendency to short lifetimes and high density, repairs can be adequately handled in the chat window. In comparison, the low density of waste information reinforces the indication given by examining lifetime, that is, that an external representation will reduce cognitive load. The low density of locations similarly indicates this.

The implication of the density score for plans is less clear; while it is quite high in comparison to wastes, it indicates that plans do not dominate conversation as strongly as repairs do. Instead, plans are interwoven with other information. This appears to indicate that users refer back and forth to other information while planning, meaning that plans, if stored in an external representation, need to be presented in a way that lets them be easily referred to multiple times as they are being revised and discussed.

Visualizing iota data Visualizing the analysis data is useful for revealing general differences between iota types. In the scatter plot shown in Figure 11, the general differences between plan, waste, and location iotas are visible. Here, the iotas from an analysis of non-CR VesselWorld groups have been plotted with one axis being the lifetime of relevance — the ratio of the number of utterances between first and last mention of the iota to the total number of utterances in the session — and the other being the number of mentions to the iota during that span. Due to the wide variance in the data, the lifetime axis is logarithmic. Also, the source data has been jiggled slightly (small fractions have been added to the discrete source values) to reveal instances where multiple data points overlap.

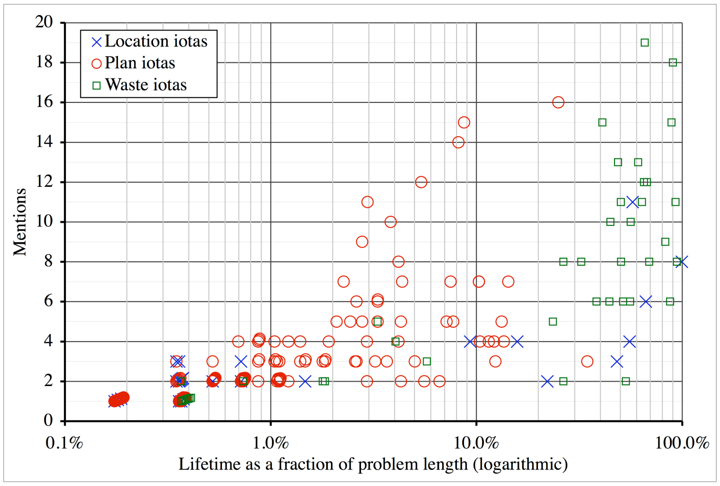


Figure 11: Differences in iota type access patterns are visible in a scatter plot

This sort of comparison graph is useful for examining the data for outliers and to check the distinctiveness of iota types. If the populations of two different iota types are very similar, the analyst may wish to examine whether they are variants of the same sort of information. The converse situation is shown in Figure 12. Only the “Location” iotas, where users have referred to a particular location in the harbor, are shown here. After plotting the Location iotas in this fashion it became clear that there were two distinct clusters of iotas. By going back and examining the source data, we discovered that the iotas in one set corresponded to those to whom only deictic references, such as “over here”, or “right there”, were made. These iotas had a much shorter lifetime than those in the other group, which corresponded to static location references such as “362,163”, or “near lbarge”.

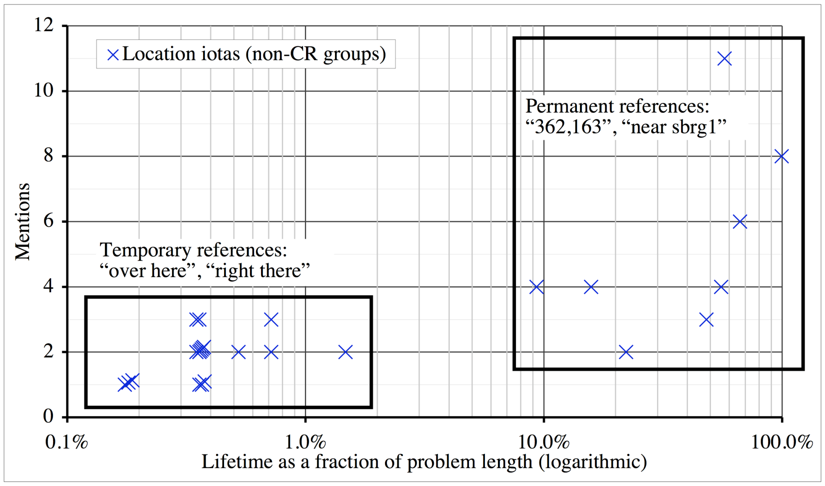


Figure 12: A scatter plot of location iotas reveals clusters of deictic vs. definite references

From this evidence it is apparent that the two types of location reference were handled quite differently by the participants. Locations that were referred to purely with deictic references tended to have a more shorter lifetime than those who were referred to by some form of definite reference. The representation provided for storing location references (the Object List) removed context from the location information encoded in it, participants did not encode deictic references in it — at a distance of time, an object list entry whose location was listed as “over here” would be difficult to connect to a particular waste. Therefore, it was suitable for location references of the second type but not for those of the first type. A redesign that included a way to address these purely deictic referents could improve performance. Such a new representation would necessarily have its own set of trade-offs.

3.4 Examining previous results

Observations of differing information access patterns gave us insight into why participants in the VW3 experiment did not make use of all available representations. In the VW3 experiment, the new coordinating representations were only partially adopted. Specifically, there were two unexpected rejections: the high-level Strategy window, and the Leaking and Action columns of the Object List. By performing a referential structure analysis on our existing data we were able to shed some light on these results.

The Strategy window provided an external representation for planning information. However, the way in which the information was presented was at odds with the way that users shared planning information in many important ways. First and foremost, our analysis above showed that users discuss plans only briefly — lifetime for a plan iota averaged twelve lines of chat, and was commonly much shorter. Because of this, it was barely worth the effort for a participant to encode the plan into the Strategy window, a task that was noticeably more difficult than simply describing the plan in chat. Another noticeable effect was that plans tended to be quickly mentioned a few times when they first appeared. This represented a discussion and negotiation of the plan itself, commonly going from a very underspecified plan to one that was understood to the satisfaction of its participants. In contrast, the Strategy window required a plan author to describe the plan definitively from the start Some users, as noted previously, felt this made the act of creating a plan in Strategy a form of authoritative planning, robbing others the opportunity to participate in negotiations about that plan. Finally, participants were usually discussing one simple plan at a time, again because of the relatively short lifetime of plans. This meant that the burden of remembering the current plan was not onerous; participants could rely on short-term memory to store this information instead of transcribing to an external representation.

The case of the unused columns in the Object List required careful reinvestigation of the waste iotas. We found that, despite the fact that waste iotas had long lifetimes and low density — implying the need for a persistent representation — particular aspects of the waste information behaved differently. Specifically, the status information meant to be stored in the “Action” column changed very frequently, and the transitions between states were either broadcast by users in the chat window as a side effect of planning, or were uninteresting to other users and hence went unshared. Hence, the effort to update the “Status” column appeared unnecessary to users, as that functionality was taken care of by other procedures they executed.

The Leaking column provided a persistent storage medium for a simple but important fact: whether or not a particular waste was leaking. However, in practice, a leaking waste dominated the activity. Wastes leaked infrequently, there was a high cost (in terms of score) of leaving a leaking waste unattended, and in almost all cases only one waste was leaking at a time; because of these factors, a leaking waste became the focus of the participants. Because of its importance, and the simplicity of the information, participants were willing and able to store the fact that a particular waste was leaking in their short-term memory. They therefore did not need a persistent, shared representation to remind them of the leaking status. Because the extra work required to transcribe and update the "leaking" information into the Object List did not provide a comparable payoff in terms of reducing cognitive load, participants felt no need to take this step.

4 General applicability of the method

Our next set of goals was to demonstrate the general applicability of our methods. To establish general utility of the method we needed to show a few important qualities of the method:

1. Representation systems have different referential structures in the discourse.
2. New analysts can be taught how to successfully apply the method.
3. Different analysts draw similar conclusions from the same data.
4. Analysts can achieve insights in a variety of domains.

This section will detail an investigation into the statistical examination of the methods and then summarize experiments that were performed to verify the latter three points.

4.1 Representation systems have characteristic information access patterns

We ran experiments with the VesselWorld data to show that differences in the discourse structure due to group membership are insignificant compared with the differences across information types and representation systems. In other words, all groups using the same representation system for the same task display similar information access patterns for each type of iota. However, groups with different representation systems have significantly different patterns of discourse, even when they are engaged in the same task. These two observations can be formulated as the following two hypotheses:

1. The variance of data within-condition (e.g., for all non-CR groups) is low: the differences between iota types (in both lifetime and density) are similar for all groups in one condition.
2. The variance of data between-conditions (i.e., non-CR groups vs. CR groups) is high: that is, the differences between iota types are significant across groups.

Validation of these two hypotheses provides important guarantees that the conclusions we are drawing from referential structure analysis are valid.

The data supported the first hypothesis. We found that different non-CR groups exhibit slight variation in how they handle a particular type of information, but these differences are minor in comparison to the differences between information types. That is, the access patterns for information do not depend on the specific group of users, but rather on the type of information. To show this statistically we carefully examined the differences between participant groups in our experiment. We chose to focus on the three most commonly occurring iota types (plans, wastes, and locations). An F-test on these iota types showed that effect due to group membership was not at all significant. The miniscule values for the F-test (e.g. for plans, $F(1, 210) < 0.5, p < 0.01$) indicate that variability between groups is much less than variability within groups. The conclusion is that it is highly unlikely that access patterns depend on which particular group of participants generates them; they must instead depend on other variables, such as the characteristics of the information itself.

The second hypothesis states that information access patterns are dependent on the representation system provided. To establish proof we performed a referential structure analysis on data from the CR groups and compared the resulting information access patterns with the non-CR groups. A T-test performed on the iota data generated for the two experimental conditions showed that it was very unlikely that the differences between the iota data for the non-CR and CR conditions were due to chance; therefore, we tentatively attribute the change in information access patterns to the change in representation system. A summary of the relevant figures appears in Figure 13.

As expected, the effect that the representation system had on iotas was dependent on iota type. Most noticeable is the strong effect on waste iotas. As explained previously, the new representation system impacted the way that users talked about waste much more heavily than they way they talked about plans and locations. This is reflected in the data; plan

T-Test	% Lifetime		Mentions	
Plans	$t(210) = 0.38,$	$p > 0.5$	$t(210) = 1.96,$	$p < 0.1$
Wastes	$t(72) = 3.61,$	$p < 0.01$	$t(72) = 2.96,$	$p < 0.01$
Locations	$t(43) = 1.48,$	$p < 0.15$	$t(43) = 1.41,$	$p < 0.2$

Figure 13: Representation system has an effects on information access patterns for certain iota types.

iotas (which were generally unaffected by the rejected introduction of the Strategy window) show little effect, whereas the statistics for waste iotas (strongly affected by the introduction of the Object List) show a very significant change. Locations, whose access patterns were changed somewhat by the new representation system, show only a moderate degree of separation. Overall, this test showed that the differences between the non-CR and CR iota data were significant.

4.2 Teaching the methods to new analysts

With the help of a class of students, we ran two experiments to gather data on how well the methodology could be taught and employed on novel domains. In this section we summarize the successes and failures of this vetting of the experimental method; a more detailed version of this analysis is available in (Feinman 2004).

In the Fall of 2003, a class composed of twenty-one Master's students and upper-level undergraduates were taught the analysis techniques presented in this paper. They applied these techniques to a set of standardized transcripts, which were used to provide feedback about the method and about how well they had learned the methods. The class was then split into groups of two to four students; each student group created problems for pairs of subjects to solve cooperatively. The groups then ran experiments and analyzed data that they generated using the methods outlined in this paper. From this analysis they were able to draw conclusions about how to alter the representation systems of their experimental applications. Most groups were able to successfully apply the methods to suggest interesting redesign possibilities for their systems.

The students were initially given a groupware system, GrewpTool, consisting of a shared editor, a textual chat, and a shared web browser (Langton, Hickey & Alterman 2004; Hickey, Langton, et al. 2004). The tool provides a shared work environment for two or

more users, including a shared text area with text color-coded by author, a chat window, and shared and private web browsers. Actions taken in the system can be replayed using a built-in VCR-like tool, allowing the application of our analysis methodologies.
Students were split into groups of two to four and were asked to design an experiment where a pair of users would employ the GrewpTool to collaboratively solve a problem. Topics ranged from "plan a 5-night vacation to Boston" to "the wedding dinner planner" to "create a web page describing the culture of a nation." The students then recruited three or four pairs of subjects, trained them in use of the system, and generated about 10 total hours of use data. From this set of data the students were asked to select a single transcript and apply the methods presented in this paper to analyze the interaction.

4.3 Analysts draw similar conclusions from the same data

The students were also asked to perform a referential structure analysis of four standard transcripts to test their analytic skills. These transcripts were pulled from data of undergraduates engaged in a pairs-programming session. Parts of this study had been discussed in class on several occasions, so while the students had not seen the specific data they were given, they were familiar with the domain. After the analyses were performed, we engaged the class in a discussion of the results and methods from this analysis, which yielded strong positive feedback about the utility of the method. In addition to providing students with unambiguous feedback about their ability to perform the analysis correctly, this exercise allowed us to test the inter-coder reliability of the methods presented here. Each transcript was analyzed by five pairs of students. The resulting analyses were qualitatively similar, though there were minor variations in results from group to group. About half the groups matched the expert analysis. Groups usually found comparable iotas and made similar conclusions, even where their analyses differed in detail. These differences can in the main be attributed to differing skill levels between student groups. The appearance of this agreement is a most encouraging sign of the applicability of the method. Again, a more detailed description of these results can be found in (Feinman 2004).

4.4 Application of the methods to a variety of domains

Students were asked to submit ideas for redesigning the GREWP tool, based on conclusions from their analysis. The students were given three weeks to generate and submit designs for new representations to improve user performance in their particular domain, with the requirement that these new designs be motivated using the analysis techniques discussed in class, including those demonstrated in this paper.

	Recurrence analysis			Referential structure analysis	
Project	Recurring coordination	Recurring errors	Secondary structure	Iota types	Iota measures
Class web page	x			x	
Collaborative coding	x	x			
Boston Adventure	x	x			
Collaborative coding	x	x			
Country web page	x	x			
Social dinner	x	x			
Trip planner	x	x	x		
Themed web page	x	x		x	x
Wedding dinner	x	x		x	x
Boston trip	x	x	x	x	x

Figure 14: Rationales offered by student groups for redesign

All of the groups of students who submitted a redesign were able to successfully motivate that redesign using these methods. As summarized in Figure 14, every group found recurring patterns of coordination and recurring errors in the interaction and used these observations to justify and shape their redesign. In some groups the students also identified the creation of secondary structure by the users. About half of the student groups were able to further refine these design ideas by pulling inferences from the referential structure analysis of their data by making assumptions based on the iota types they identified. Most of these groups employed the full method, computing and comparing various measures (such as iota lifetimes and density of mentions) derived from their data. In the next few sections we will examine these results in greater detail.

4.4.1 Using recurrence analysis for redesign

Recurring coordination Looking at the rationales for redesign presented by the students (Figure 14), we see that all ten groups were able to identify recurrent patterns of coordination in the data sufficient to warrant a redesign. In addition, all but one group used the appearance of recurrent errors in their data to justify the necessity for a new representation system. Both of these results are very encouraging indicators of the usefulness of this form of analysis. The recurring situations identified centered around the heart of the interaction in each case. For example, in the “wedding planner” system, the students noted users spent a great deal of time discussing seating arrangements. Coupled with other observations this led them to create representations for coming up with seating charts.

Recurring errors Almost all groups used the appearance of recurring errors as design justifications. For the student groups, this indicator provided some of the richest data. Despite the overall paucity of data, users made many mistakes that indicated that the representation system required improvement. For example, the subjects in one group were asked to plan a road trip from Boston to Los Angeles. They often made errors related to problems with attention; that is, one user would enter something into the shared text area, but the other user would fail to notice, and instead duplicate the efforts of the first user. As a result the designers proposed a representation that would allow users to keep track of what task each user was working on.

Secondary structure Use of appearance of secondary structure in the data was less frequently investigated by the students — only two groups justified their redesign based on the appearance of such structure. This is in accordance with our expectations. Because of the relatively small data set collected by the students — only ten hours of data, with each group only using the tool for a few hours — there is little time for the subjects to generate useful secondary structure. In addition, significant sophistication on the part of the analyst is required to spot small-scale, procedure structure such as adjacency pairs.

The structure found by the students is nevertheless compelling. For example, in the “Boston trip” group, one of the subjects ended up filling the shared text editor pane with a highly-formatted itinerary. The subjects felt the need to create a shared representation to organize their activity; however, the tools at their disposal were minimal — only shared text editor — and so they were unable to generate a truly effective representation. The redesign for this domain addressed this and other problems by including a tabular shared itinerary representation similar to the Object List.

4.4.2 Using referential structure analysis for redesign

The students made a slightly different use of the referential structure analysis than anticipated. Only half of the groups made use of the referential structure analysis in justifying their redesign. However, all of these groups used the regimen of identifying new iota types as a way to discover the most important topics for discussion in their domain. Armed with this knowledge they produced designs that incorporated shared, structured external representations for these kinds of information. These new iota types and new representations are summarized in Figure 15.

Project domain	New iota types	New representations
Class web page	webpage	Browser history
Boston Trip	event location price	To-do list Itinerary Budget calculator
Themed web page	requirement topic	Requirement list Topic list
Wedding dinner	constraint food guest	Seating Chart Menu Planner Guest List
Trip planner	event time	Timeline

Figure 15: Students designed new representations based on finding new iota types

Only three groups actually drew conclusions based on the statistical analysis of iota data – i.e., lifetime, density, and so forth. We attribute this to a number of causes. Most importantly, the students were only required to perform full referential structure analyses on a subset of their complete data, and so had a relatively small data set from which to draw conclusions. Hence, whatever data they did have was likely quite noisy, making it hard to draw conclusions from. In addition to this, students who were able to come up with a plausible redesign using the easier methods shown above were unlikely to then continue on to perform a detailed analysis of iota access patterns. This was likely due both to time constraints and to the relative simplicity of the domains being investigated.

The groups that did perform the full analysis were able to focus their attention on the more important iota types, and were also able to design representation systems that more closely matched the access patterns of the information they encoded. For example, the “wedding dinner” group examined closely the conversations their users were having while planning the (theoretical) dinner. They found exchanges about budget to be a frequent occurrence, with many brief mentions of what they termed the budget “constraint” iota. From these insights they were able to design a shared representation — a budget calculator — that they felt matched the access characteristics of their data.

5 Conclusions

Theoretical frameworks such as distributed cognition and activity theory base their analysis on observation of participants at work. However, the methods these frameworks provide for interpreting collected data are still fairly abstract and require a high degree of sophistication on the part of the analyst. They generally do not establish particular methods that an analyst can use to apply interpretations of these observations. Because of this top-down approach, these methodologies are well-suited for abstract conclusions but can be difficult to use to extract concrete design decisions.

Conversation and discourse analysis provide techniques for examining transcribed conversation. The utterance-by-utterance analysis of communication that these bottom-up techniques provide gives insights that the theoretical frameworks do not. These techniques excel at examining the implications of utterance-level features such as word and speaker choice. However, these techniques do not explicitly deal with complexities such as long-term interactions, the cognitive load of participants, and the representation of information.

Our approach combines methods and ideas from conversation analysis and distributed cognition to create two analysis techniques. The first technique focuses on recurring interaction between participants, and the secondary structure they create in response to recurring problems. The trouble spots in the coordination, once established, provide an analyst with starting points for a redesign effort. The second technique examines the references that participants make in communicating with each other to progress in their activity and consequently the way information is passed around and stored in representations. By examining these references we can see what sorts of information are being shared, and how the flow of information is mediated by the representation system. Based on these observations an analyst can draw conclusions about how specifically to improve the existing representation system to better suit the flow of information. The analyst is given insight into a number of specific opportunities for redesign: areas where introducing a mediating artifact could improve the interaction, the types of information that need to be supported in the representation system, and the way representations should be designed to store and present that information.

We have shown that together these techniques provide a framework that:

1. can be applied to a wide variety of domains
2. can easily and quickly be taught to other analysts
3. produces similar results when applied by different analysts to the same interaction
4. provides a strong theoretical framework for looking at low-level interaction data
5. produces quantitative data to support conclusions about the interaction
6. offers concrete recommendations for redesign

The first experiment, comparing performance of users in two versions of Vesselworld, demonstrated the effect of alterations to the representation system. We were able to establish the utility of recurrence analysis by demonstrating its usefulness in designing the new representation system, and explain the choices of representation that users made based on a referential structure analysis. We presented visualizationtools that allow the analyst to explore the data and discover unexpected findings. Data also showed that different groups with the same representation system had similar patterns of discourse.

The second experiment, where students were asked to apply the methodology, showed that the method could successfully be taught and applied to a variety of domains. These students also provided data which demonstrated that different groups achieved similar results when applying the methods. This proved that the methods could be taught, applied, and used in a reproducible manner to generate effective redesign ideas for groupware applications. The methodology provided the students with a step-by-step procedure to use to refine their applications, giving them guidance as to what portions of the interaction to address.

Our results show that the analytic methods presented provide a useful way to look at interaction data. The goal of our analysis is to inform a redesign of the representation system and procedures surrounding the representations, in order to enhance the performance of participants. To this end we are continuing in our refinement of these methods. We are working toward providing a clear connection between the results of these analyses and concrete suggestions for redesign.

NOTES

Acknowledgments. Thanks goes to Seth Landsman, Josh Introne, Alex Bradley, and Dan Ritter for comments and continued collaboration in support of this paper. Special thanks also goes to the Fall 2003 class of CoSi111a for providing extensive feedback on these methods.

Support. This research was supported by the Office of Naval Research under grants No. N00014-96-1-0440 and N66001-00-1-8965.

REFERENCES

van der Aalst, W., ter Hofstede, A., Kiepuszewski, B., & Barros, A. (2003). Workflow patterns. Distributed and Parallel Databases, 14(3), 5-51.

Alterman, R., & Garland, A. (2001). Convention in joint activity. Cognitive Science, 25, 4.

Alterman, R., Feinman, A., Introne, J., & Landsman, S. (2001). Coordinating representations in computer-mediated joint activities. Proceedings of 23rd Annual Conference of the Cognitive Science Society. Hillsdale, NJ: Lawrence Erlbaum Associates.

Apple Computer. (1987). Human interface guidelines: the Apple desktop interface. San Francisco: Addison-Wesley.

Bødker, S. (1990). Through the interface — a human activity approach to user interface design. Hillsdale, NJ: Lawrence Erlbaum Associates.

Borchers, J. (2001). A pattern approach to interaction design. New York: John Wiley & Sons.

Card, S., Moran, T., & Newell, A. (1983). The psychology of human-computer interaction. Hillsdale, NJ: Erlbaum.

Carroll, J. (Ed.) (2003). HCI models, theories, and frameworks: toward a multidisciplinary science. Boston: Morgan Kaufman Publishers.

Clancey, W. (2002) Simulating activities: relating motives, deliberation, and attentive coordination. Cognitive Systems Research, 3(3) 471-499.

Clark, H. (1996). Using language. New York: Cambridge University Press.

Clark, H., & Brennan, S. (1991). Grounding in communication. In J. Levine, L.B. Resnik, & S.D. Teasley (Eds.), Perspectives on Socially Shared Cognition (127-149). New York: American Psychological Association.

Clark, H. & Wilkes-Gibbs, D. (1986). Referring as a collaborative process. Cognition, 22, 1-39.

Cole, M., & Engeström, Y. (1993). A cultural-historical approach to distributed cognition. In G. Solomon (Ed.), Distributed Cognitions: Psychological and Educational Considerations, pp. 1–46. New York: Cambridge University Press.

Carey, M., Stammers, R., & Astley, J. (1989). Human-computer interaction design: the potential and pitfalls of hierarchical task analysis. In D. Diaper (Ed.), Task analysis for human-computer interaction, 56-74. Chichester, UK: Ellis Horwood.

Drew, P., Chatwin, J., & Collins, S. (2001). Conversation analysis: a method for research into interactions between patients and health-care professionals. Health Expectations, 4 (1).

Drew, P., & Heritage, J. (1992). Talk at work. New York: Cambridge University Press.

Ellis, C., Gibbs, S., & Rein, G. (1991). Groupware: some issues and experiences. Communications of the ACM, 34, pages 38-58.

Engeström, Y., Miettinen, R., & Punamaki, R. (Eds.). (1999). Perspectives on activity theory. New York: Cambridge University Press.

Feinman, A. (2004). From conversation analysis to groupware design. Ph.D. Thesis. Waltham, MA: Brandeis University. (Forthcoming)

Feinman, A., & Alterman, R. (2003). Discourse analysis techniques for modeling group interaction. Proceedings of the Ninth International Conference on User Modeling, 228-237. New York: Springer-Verlag.

Fitzpatrick, G. (2003). The locales framework: understanding and designing for wicked problems. Norwell, MA: Kluwer Academic Publishers.

Foster, G., & Stefik, M. (1986) Cognoter: theory and practice of a colaborative tool. In Proceedings of the 1986 ACM conference on Computer-supported cooperative work. Austin, Texas: ACM Press, 7-15.

Fjeld, M., Lauche, K., Bichsel, M., Voorhorst, F., Krueger, H., & Rauterberg, M. (2002). Physical and virtual tools: activity theory applied to the design of groupware. In Nardi, B. & Redmiles, D. (Eds.) A Special Issue of Computer Supported Cooperative Work (CSCW): Activity Theory and the Practice of Design, Volume 11 (1-2), 153-180.

Garfinkel, H. (1967). Studies in ethnomethodology. Englewood Cliffs, NJ: Prentice-Hall.

Gould, J., & Lewis, C. (1985). Designing for usability: key principles and what designers think. Communications of the ACM, 28 (30), 300-311.

Hickey, T., Langton, J., Granville, K., & Alterman, R. (2004) Enhancing CS programming lab courses using collaborative editors. To be presented at CCSCE04 (15-16 Oct 2004).

Langton, J., Hickey, T., & Alterman, R. (2004). Integrating tools and resources: a case study in building educational groupware for collaborative programming. The Journal of Computing Sciences in Colleges, 19(5), 140-153.

Hollan, J., Hutchins, E., & Kirsh, D. (2000). Distributed cognition: toward a new foundation for humancomputer interaction research. ACM Transactions on Computer-Human Interaction, 7(2), 174-193.

Hutchins, E. (1995a). How a cockpit remembers its speeds. Cognitive Science, 19, 265-288.

Hutchins, E. (1995b). Cognition in the wild. Cambridge, MA: MIT Press.

John, B., & Kieras, D. (1996). The GOMS family of user interface analysis techniques: comparison and contrast. ACM Transactions on Computer-Human Interaction, 3, 320-351.

John, B., & Kieras, D. (1996). Using GOMS for user interface design and evaluation: which technique? ACM Transactions on Computer-Human Interaction, 3 (4), 287-319.

Kirwan, B. & Ainsworth, L.K. (Eds.) (1992). A Guide to Task Analysis. London: Taylor and Francis.

Landsman, S., & Alterman, R. (2003). Building groupware on THYME. (Technical Report CS-03-234). Waltham, MA: Brandeis University.

Landsman, S., & Alterman, R. (2002). Analyzing usage of groupware. (Technical Report CS-02-230). Waltham, MA: Brandeis University.

Leont'ev, A. (1978). Activity, consciousness, and personality. Englewood Cliffs, NJ: Prentice-Hall.

Lewis, D. (1969). Convention: a philosophical study. Cambridge, MA: Harvard University Press.

Lockman, A. and A.D. Klappholz (1978). Toward a procedural model of contextual reference solution. Discourse processes, 3, 25-71

Malone, T., Grant, K., Lai, K., Rao, R., & Rosenblitt, D. (1987). Semi-structured messages are surprisingly useful for computer-supported coordination. ACM Transactions on Office Information Systems, 5 (2), 115-131.

McCabe, R., Heath, C., Burns, T., & Priebe, S. (2002). Engagement of patients with psychosis in the consultation: conversation analytic study. British Medical Journal. 325 (7373), 11481151.

Mueller, M., Ganslandt, T., Frankewitsch, T., Krieglstein, C., Senninger, N., & Prokosch, H. (1999). Workflow analysis and evidence-based medicine: towards integration of knowledge-based functions in hospital information systems. Poster at GMDS Jahrestagung 1999.

Nardi, B., Whittaker, S., & Bradner, E. (2000). Interaction and outeraction: instant messaging in action. In Proceedings of the Conference on Computer Supported Cooperative Work (CSCW 2000). Philadelphia, PA.

Nielsen, J. (2000). Designing web usability: the practice of simplicity. Indianapolis, IN: New Riders Publishing.

Norman, D. (1986) The psychology of everyday things. Hillsdale, NJ: Lawrence Erlbaum Associates.

Norman, D. (1986). Design principles for human-computer interfaces. In Berger, D., Pezdek, K., & Banks, W. (Eds.). Applications of cognitive psychology: Problem solving, education, and computing. Hillsdale, NJ: Lawrence Erlbaum Associates.

Norman, D. (1991). Cognitive artifacts. In J. M. Carroll (Ed.), Designing interaction: Psychology at the human-computer interface. New York: Cambridge University Press.

Norman, D. (1998). The invisible computer: why good products can fail, the personal computer is so complex, and information appliances are the solution. Cambridge, MA: MIT Press

Perkins, D. N. (1993). Person-plus: a distributed view of thinking and learning. In Salomon, G. (Ed.), Distributed cognitions: Psychological and educational considerations, 88-110. New York: Cambridge University Press.

Rasmussen, J. (1986). Information processing and human-machine interaction: an approach to cognitive engineering. New York: North-Holland.

Rittel, H., & M. Webber (1984). Planning problems are wicked problems. In Cross, N. (Ed.), Developments in Design Methodology, pp. 135-144. New York: John Wiley & Sons.

Sacks, H. (1992). Lectures on conversation. Oxford: Basil Blackwell.

Sacks, H., Schegloff, E., & Jefferson, G. (1974). A simplest systematics for the organisation of turn-taking for conversation. Language, 50, 696-735.

Schegloff, E., & Sacks, H. (1973). Opening up closings. Semiotica, 7, 289-327.

Schegloff, E. (1979). Identification and recognition in telephone conversation openings. In Psathas, G. (Ed.), Everyday language studies in ethnomethodology (pp. 23-78), New York: Irvington Publishers.

Schegloff, E. (1980). Preliminaries to preliminaries: ‘Can I ask you a question?’. Sociological inquiry 50 (3-4), 104-152.

Schegloff, E. (1991). Conversation analysis and socially shared cognition. In Levine, J., Resnik, L.B., & Teasley, S.D., (Eds.), Perspectives on socially shared cognition, 150-171. New York: American Psychological Association.

Schmidt, K. & Bannon, L. (1992). Taking CSCW seriously: supporting articulation work. Computer Supported Cooperative Work (CSCW), 1 (1-2), 7-40.

Schmidt, K., & Wagner, I. (2002). Coordinative artifacts in architectural practice. In Blay-Fornarino, M., et al. (Eds.): Cooperative systems design: a challenge of the mobility age, Proceedings of the Fifth International Conference on the Design of Cooperative Systems (COOP 2002), 257-274.

Schneiderman, B. (1998). Designing the user interface: strategies for effective human-computer interaction. Reading, MA: Addison-Wesley.

Searle, J. (1969). Speech acts: an essay in the philosophy of language. New York: Cambridge University Press.

Schraagen, J., Chipman, S., & Shalin, V. (Eds.) (2000). Cognitive task analysis. Mahwah, N.J.: Lawrence Erlbaum Associates.

Sierhuis, M, Clancey, W., & van Hoof, R. (2003). Brahms: a multiagent modeling environment for simulating social phenomena. Presented at the first conference of the European Social Simulation Association (SIMSOC VI), Groningen, The Netherlands.

Slembrouck, S. (2003). What is meant by discourse analysis? Retrieved July 20th, 2004, from http://bank.rug.ac.be/da/da.htm

Suchman, L. & Trigg, R. (1991). Understanding practice: video as a medium for reflection and design. In Greenbaum, J., & Kyng, M. (Eds.), Design at Work, 65-90, Hillsdale, N.J.: Erlbaum.

Suchman, L., & Trigg, R. (1993) Artificial intelligence as craftwork. In Chaiklin, S. and Lave, J. (Eds.), Understanding Practice: Perspectives on Activity and Context, 144-178. New York: Cambridge University Press.

Strauss, A. (1985). Work and the division of labor. The Sociological Quarterly, 26 (1), 1-19.

Stubbs, M. (1983). Discourse analysis: the sociolinguistic analysis of natural language. Oxford: Basil Blackwell.

Tannen, D. (1990). Gender differences in topical coherence: creating involvement in best friends' talk. Discourse Processes 13 (1), 73-90.

Taylor, F.W. (1911). The principles of scientific management. New York: Harper Bros.

Vicente, K. (1999). Cognitive work analysis: toward safe, productive, and healthy computer-based work. Mahwah, NJ: Lawrence Erlbaum & Associates.

Vygotsky, L. (1962). Thought and language. Boston, MIT Press.

Whittaker, S. & Sidner, C. (1996). Email overload: exploring personal information management of email. In Proceedings of CHI'96 Conference on Computer Human Interaction, NY: ACM Press, 276-283